国外电子与通信教材系列

超大规模集成电路与系统导论

Introduction to VLSI Circuits and Systems

[美] John P. Uyemura 著

周润德 译

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

VISIT…

LANZAROTE Caliente.com

超大规模集成电路与系统导论

Introduction to VLSI Circuits and Systems

一本全面介绍现代数字 VLSI 设计的导论书

美国乔治亚理工学院（Georgia Institute of Technology）的 John P. Uyemura 教授在本书中以清晰的语言和丰富的图表全面介绍了现代 VLSI 设计概念，使这些概念很容易为许多领域（如电气工程、计算机工程和计算机科学）的学生和工程技术人员所掌握。本书把CMOS逻辑电路、集成电路制造和版图设计有机地结合在一起，并且上升到系统层次考虑。

全书着重介绍设计复杂 VLSI 芯片所采用的自顶向下的设计过程，使读者清楚地了解在层次化设计中如何对每一层次进行全面综合的考虑，学习如何比较一个CMOS逻辑问题的不同解决方案，以及选择不同方案的结果。

本书特点：

- 既适合课堂教学也适合自学
- 全面介绍现代 VLSI 设计概念
- 以独立的章节介绍 Verilog® HDL
- 用大量示例说明系统层次的设计思路
- 习题丰富，有助理解关键概念
- 详细阐述系统层次设计与较低层次中所考虑的开关速度和芯片面积之间的相互作用
- 涵盖高速 CMOS 逻辑电路设计

CD-ROM 中包含 Verilog® 学生版和 SPICE 模拟器

ISBN 7-5053-9424-X
9 787505 394247 >

责任编辑：杨丽娟
特约编辑：明足群
封面设计：毛惠庚

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书

ISBN 7-5053-9424-X　定价：56.00 元（含光盘一张）

国外电子与通信教材系列

超大规模集成电路与系统导论

Introduction to VLSI Circuits and Systems

【美】 John P. Uyemura 著

周润德 译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书介绍CMOS数字大规模集成电路与系统设计的基础。全书分为三部分。第1部分介绍集成电路的逻辑与物理层设计，其中包括CMOS静态门的逻辑设计与信号控制、芯片生产与制造工艺，版图设计与CAD工具。第2部分讨论CMOS电子电路，介绍MOSFET的特性和开关模型，各类逻辑电路，包括高速CMOS逻辑电路，同时介绍分析逻辑链延时的经典方法和新方法。第3部分为VLSI的系统设计，介绍Verilog® HDL高层次描述语言，分析数字系统单元库部件以及加法器和乘法器的设计，并且研究物理设计中应当考虑的问题，包括时钟技术、布局布线、信号串扰、测试与功耗问题。

本书可作为电子、电气、自动化与计算机等专业本科高年级学生及研究生课程的教科书，也可作为相关科技和工程技术人员的参考书。

Copyright © 2002 John Wiley & Sons, Inc.

All rights reserved. Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书简体中文专有翻译出版权由 John Wiley & Sons Inc. 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字：01-2003-0588

图书在版编目(CIP)数据

超大规模集成电路与系统导论/(美)尤耶缪拉(Uyemura, J. P.)著；周润德译.
北京：电子工业出版社，2004.1
(国外电子与通信教材系列)
书名原文：Introduction to VLSI Circuits and Systems
ISBN 7-5053-9424-X

Ⅰ.超… Ⅱ.①尤…②周… Ⅲ.①超大规律集成电路—电路设计—教材②超大规模集成电路—系统设计—教材 Ⅳ.TN470.2

中国版本图书馆 CIP 数据核字(2003)第111120号

责任编辑：杨丽娟　特约编辑：明足群
印　　刷：北京四季青印刷厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编 100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：31　字数：794千字
印　　次：2004年1月第1次印刷
定　　价：56.00元　(含光盘一张)

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个"国外电子与通信教材系列"项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝"国外电子与通信教材系列"项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿

中国工程院院士、清华大学教授

"国外电子与通信教材系列"出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套“国外计算机科学教材系列”，在全国高校以及科研单位中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于“十五”期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进“国外电子与通信教材系列”，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，“国外电子与通信教材系列”的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过“教育部高等学校电子信息科学与工程类专业教学指导委员会”的审核，并得到教育部高等教育司的批准，纳入了“教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书”。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严谨与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

主　任	吴佑寿	中国工程院院士、清华大学教授
副主任	林金桐	北京邮电大学校长、教授、博士生导师
	杨千里	总参通信部副部长、中国电子学会会士、副理事长 中国通信学会常务理事
委　员	林孝康	清华大学教授、博士生导师、电子工程系副主任、通信与微波研究所所长 教育部电子信息科学与工程类专业教学指导委员会委员
	徐安士	北京大学教授、博士生导师、电子学系副主任 教育部电子信息与电气学科教学指导委员会委员
	樊昌信	西安电子科技大学教授、博士生导师 中国通信学会理事、IEEE 会士
	程时昕	东南大学教授、博士生导师 移动通信国家重点实验室主任
	郁道银	天津大学副校长、教授、博士生导师 教育部电子信息科学与工程类专业教学指导委员会委员
	阮秋琦	北方交通大学教授、博士生导师 计算机与信息技术学院院长、信息科学研究所所长
	张晓林	北京航空航天大学教授、博士生导师、电子工程系主任 教育部电子信息科学与电气信息类基础课程教学指导委员会委员
	郑宝玉	南京邮电学院副院长、教授、博士生导师 教育部电子信息与电气学科教学指导委员会委员
	朱世华	西安交通大学教授、博士生导师、电子与信息工程学院院长 教育部电子信息科学与工程类专业教学指导委员会委员
	彭启琮	电子科技大学教授、博士生导师、通信与信息工程学院院长 教育部电子信息科学与电气信息类基础课程教学指导委员会委员
	徐重阳	华中科技大学教授、博士生导师、电子科学与技术系主任 教育部电子信息科学与工程类专业教学指导委员会委员
	毛军发	上海交通大学教授、博士生导师、电子信息学院副院长 教育部电子信息与电气学科教学指导委员会委员
	赵尔沅	北京邮电大学教授、教材建设委员会主任
	钟允若	原邮电科学研究院副院长、总工程师
	刘　彩	中国通信学会副理事长、秘书长
	杜振民	电子工业出版社副社长

译　者　序

VLSI(超大规模集成电路)的快速发展是20世纪70年代以来人类科学历史上的一个奇迹。它是计算机和通信领域的支柱,已经成为其他科学和工程技术的基础。现在VLSI的设计和应用已经渗透到各个工程技术领域,几乎每个工程学科的学生和工程技术人员都要了解VLSI方面的知识,因此十分需要一本有关VLSI电路与系统设计,能够适用于各类工程专业的教科书和自学教材。美国乔治亚理工学院(Georgia Institute of Technology)的John P. Uyemura教授2002年所著的《超大规模集成电路与系统导论》(Introduction to VLSI Circuits and Systems)就是这样一本优秀的教材。

John P. Uyemura教授在美国Georgia Institute of Technology长期从事VLSI领域的研究和教学,有很深的学术造诣和丰富的教学经验,在本书出版之前已经出版了多本这一领域的教科书。本书则是他最近的又一力作,内容深入浅出,从介绍最基础的知识开始,逐步讨论深层次的专题,每章的最后列出许多很有价值的参考资料。因此本书不仅可以作为电子、电气、自动化与计算机等专业本科高年级学生及研究生课程的教科书,也可作为相关科技和工程技术人员的参考书。

本书共16章。除了第1章VLSI概论以外,其余分为三部分。第1部分从介绍CMOS技术的逻辑设计开始,逐步进入物理层次,其中包括芯片生产与制造工艺,版图设计和CAD工具。第2部分讨论CMOS电子电路,介绍MOSFET的特性和开关模型,各类逻辑电路,包括高速CMOS逻辑电路,同时介绍分析逻辑链延时的经典方法和新方法。第3部分为VLSI的系统设计,介绍Verilog® HDL高层次描述语言,分析数字系统单元库部件以及加法器和乘法器的设计,并且研究物理设计中应当考虑的问题,包括时钟技术、布局布线、信号串扰、测试与功耗问题。

本书在翻译过程中得到了电子工业出版社的热心指导和大力支持,得到了清华大学微电子学研究所领导和多位教师的关心,特别是得到了朱钧教授、贺祥庆教授、吴行军副教授等多位老师的帮助与指正。我的博士研究生戴宏宇、张盛、王乃龙、杨骞、肖勇、张建良等在完成译稿过程中给予了我很大的支持。我的妻子金申美和女儿周晔不仅帮助翻译和修改了部分章节,而且完成了全部的文字输入和文稿整理。在此一并深表谢意。

最后,本书虽经仔细校对,但由于译者水平有限,文中定有不当或欠妥之处,望读者批评指正。

译　者

2003年10月于清华园

前　言

VLSI 已经成为现代技术的一个主要推动力。该领域是计算和通信的基础，并且仍在以神奇的步伐继续快速发展。

本书作为一本教科书涵盖数字 CMOS VLSI 系统设计的基础。它可用于电气工程、计算机工程和计算机科学领域高年级大学生的第一门课程。在补充杂志文献后，也可用于一年级研究生的课程。本书的核心内容可以用 10 周的课时讲完，但是全部教材足够 15 周的课时使用，或者加上最近文献的阅读可以作为两个学期(quarter 制)的教材。

学习本书不需要 VLSI 或 CMOS 的背景知识，但应该学过电气或计算机工程的核心课程，其中最重要的是数字逻辑设计和电路分析。本书电子学部分从最基本的概念开始，所以计算机科学专业的学生应当能够跟上讨论。本书是 VLSI 领域的入门书，并不要求预先了解这一领域。它也适用于学生和专业人员自学。虽然讨论是从最基础开始的，但也涉及高水平课题的深层次分析。

内容介绍

在**第 1 章**简要介绍设计层次化的一些术语和概念后，教材分为三个主要部分。

第 1 部分——硅片逻辑。介绍 CMOS 技术中逻辑设计的概念。它引导学生从逻辑设计概念进入 CMOS 电路设计。**第 2 章**介绍作为简单逻辑控制开关的 MOSFET，然后集中于布尔逻辑层次上的 CMOS 静态逻辑门设计。**第 3 章**初步了解物理设计，将集成电路看做用来控制信号流的一组图形材料层。这使我们从开关级的描述向下进入物理层。**第 4 章**覆盖硅芯片生产的内容，考察生产工艺的一般和特殊方面。**第 5 章**是本部分讨论的最高点，其中考察了物理设计和版图的细节问题，对设计体系和单元库的讨论将这些材料与 CAD 工具的使用联系起来。第 1 部分学习结束后，学生就能进行 CMOS 的基本逻辑设计并了解它们是如何转换到硅环境中去的。

第 2 部分是 CMOS 的电路部分——从逻辑到电路。它侧重于系统设计层的简单含义和开关模型。**第 6 章**用平方定律公式说明 MOSFET 的特征，用来建立线性 RC 开关模型。**第 7 章**覆盖 CMOS 逻辑电路的电气特性。**第 8 章**介绍高速 CMOS 链，讨论经典方法，但也包括 Logical Effort(逻辑努力)一节。**第 9 章**考察一些高级 CMOS 逻辑系列，内容比较专业化，适合于某些读者。

余下的章节包含在第 3 部分中——VLSI 系统设计。材料的选择将逻辑电路、物理版图以及电子电路联系在一起，构成 VLSI 系统设计的单一学科。**第 10 章**介绍 Verilog® HDL，本书选用它作为在某些部分进行高层次描述的基础；已经熟习 VHDL 的学生能够很快地掌握 Verilog。**第 11 章**把高层次 HDL 描述与逻辑、电路和物理实现联系在一起，从这个角度分析一些数字单元库部件；采用这一方法的目的是要说明如何写高层次的抽象描述，然后考察在 CMOS

VLSI 中用来实际构成部件的各种选择。同样的思想也用于**第 12 章**中介绍加法和乘法电路。这两章都强调在层次化设计中各层次间的相互作用。**第 13 章**包含存储电路及其组织结构。**第 14 章**是芯片级物理设计的一些考虑，讨论互连线模型、串扰、布局以及布线等问题。同时对电源分配技术，I/O 电路和低功耗设计做了介绍。**第 15 章**有关系统级设计，介绍时钟驱动器和分配树以及像流水线和位片设计这样的系统级数据通路技术。本书以简单介绍可靠性和测试的**第 16 章**作为结束。

作者致力于展现 VLSI 是一门交叉学科，它依赖于许许多多的专家在各个层次上的共同工作。重点放在说明不同观点间的相互作用，以及在一个层次上所做的设计选择，如何对其他所有的层次产生影响。

本书的宗旨

"VLSI"对不同的人意味着不同的含义。它由许多不同的专业组成，它们之间以一种独特的方式相互作用。一个真正的"VLSI 设计者"应该是在某个特定领域工作，但要了解各单元是如何相互作用，形成一个系统的。我在 Georgia Tech 大学教过好几年本科和研究生的 VLSI 课程。这里介绍的观点是我发现对引导各种背景基础的学生进入该领域非常有效的观点。

对于每个专题都可以在其他资料中找到更为详细的讨论。本教科书与专著的差别在于介绍的角度、深浅和应用。它为学习 VLSI 的主要方面提供一个基本统一的基础。各专题的选择基于它们对全面了解该领域的重要性。所进行的分析对推导有用的结果已经足够详细，当然要作为专家又当别论。

选择在本领域工作的学生应当集中学习一到两个领域。例如作者本人教授的高年级本科生和研究生水平的数字电子电路课程远比这本教材要深得多。在每章的后面都列有参考资料以供对各专题更深入的学习。没有列出杂志文章，考虑这不是一本专著。

教材的使用

每章的内容相当于 3 至 6 小时的授课。具体每一节或一章所需要的时间取决于学生的基础背景。大部分章后面都附有习题，对采用本教材的授课教师可以提供习题答案。本书 URL 的网址是

www.wiley.com/college/Uyemura

在本书中有些章的内容强调实际设计和版图设计，很难给出"家庭作业"式的习题。建议这些内容最好作为设计项目。

各专题的顺序安排按由下向上的设计层次介绍概念(第 1 和第 2 部分)，然后进入第 3 部分的系统层。这比较适合于电气工程和硬件方向的计算机工程的学生。对于面向系统的设计课程，从第 10 章的 Verilog 讨论开始可能会觉得更自然些。这样可以强调高层次的系统设计和抽象，而把物理设计和电子电路作为实现的基础，使本教材更适合于计算机科学和面向系统的计算机工程专业。

围绕这本书已经制定了几种课程教学大纲。这里列出高年级一学期课程的三种安排，每种都给出按章讲授顺序的建议。根据学生的基础和授课教师的兴趣可做必要的改动。第 1 部分应当按顺序讲授，其中第 3 至第 5 章可以进行得较快。第 2 部分由于是电子电路的内容需要较多的时间。其中第 6 和第 7 章对本书其余部分比较重要；第 8 和第 9 章侧重于高级电子

电路设计,可供选用。第3部分的内容比较多样。第10章关于Verilog的内容可以用3至4课时讲完。其余各章侧重于专门的内容,不必按顺序进行。第14和第15章在系统设计课程中应当包括。

一般介绍	系统课程	电路专长
第一部分	第三部分	第一部分
第2~5章	第10,15章	第2~5章
第二部分	第一部分	第二部分
第6~8章	第2~5章	第6~9章
第三部分	第二部分	第三部分
第10,11,14~15章	第6,7章	第12~15章
	第三部分	
	第14,16章	

CAD工具

对CAD工具做了一般的讨论,而没有去讨论具体的工具。所以应当注意本书所附带的CD盘,它包括具有有限功能版本的AIM-SPICE和MiroCap6 SPICE电路分析程序,以及Silos Ⅲ Verilog仿真器。这些可使许多PC用户亲身体验电路模拟和Verilog编码。CD盘当前的内容列在一个叫README.TET的文件中,文件中包含装载提示。

设计项目

设计一个项目是在课堂环境中学习VLSI的良好手段。在Georgia Tech大学最初的实验安排是有关版图,DRC,参数提取和电路模拟。这些被存放在库中,以便最后在设计规模较大的芯片时使用。由于工艺的快速发展,设计说明要求每年都要改变。

设计项目本身总是按小组完成的,每个组由5到20个学生组成,设计项目由教师布置。做过的一些项目有基本的微处理器,MIMD阵列处理器,通信接口,流水线数据通路,以及DSP电路。

项目的布置只包含一个系统层次的说明,细节完全留给学生工程小组来完成。他们必须建立一个高层次的HDL模型,用以转换为下层的行为、逻辑、电路和版图。他们拥有完整的CAD工具包,但没有综合工具。这使小组只能去使用他们在早期实验作业中设计的单元库。设计项目的意图是尽可能向学生展示VLSI的各个方面。

致谢

John Wiley & Sons公司的Mark Berrafato和Charity Robey在本书著作的早期给予我许多启示。我的编辑Bill Zobrist从未对我将最终完成此书失去过信心。甚至在他第一个孩子Ian出生之后仍然继续他的支持,这多少要在他的家庭和他的工作之间合理分配一下时间!Jenny Welter和Susannah Barr没有忘记每个细节,无论大的还是小的,使本书的出版得以顺利进行。Maddy Lesure除为本书设计封面之外还帮助我解决图表问题,使之看起来更好。最后我要向我的出版编辑Christin Cervoni致敬,她为协调本书的出版做了大量的工作,检查了每个细节!

几位审稿人对本书最后的完成提出有益的建议。在此特别要感谢 Krishnendu chakrabarty(Duke 大学),Mona Zaghloul(George Washington 大学),Ralph Teeing-Cummings (The Johns Hopkins 大学)以及 Giovanni De Micheli (Stanford 大学)各位教授。我还要感谢 Georgia Tech 大学在过去几年中上过我芯片设计课的数百名 ECE 学生。他们在设计项目上所花费的无数时间使我了解到授课的内容怎样才能更好地转化为实际的应用。他们对课程和讲义稿的反馈意见极大地帮助了我的再次修改。尤其是 Michael Robinson 特别仔细地阅读了几章。Tony Alvarez (Cypress Semiconductor 半导体公司)和 Brian Butka (IDT 公司)设法为本书提供了几幅芯片照片。

在此,我还要感谢 Georgia Tech 电学与计算机工程学院院长 Roger Webb 博士对我写这本书的一贯支持。Bill Sayle 和 Joe Hughes 教授总是设法满足我的教学要求,使我能够得到涉足 VLSI 的机会。与 John Buck 和 Glenn Smith 教授的交谈总能给我们动力和启迪。

最后,我要再次感谢我的夫人 Melba 和我的女儿 Christine 和 Valerie,感谢她们对我的这项(以及所有的)工作的无限耐心和支持。虽然我不能够回报给她们我在写这本书上所花费的时间,也许去法国做一次短期旅行可以有所补偿。

再见!

John P. Uyemura
Atlanta, GA
2001 年 4 月

国外电子与通信教材系列 ①

类别	中文书名	英文书名	作者	译审者	版别
电路理论与应用	电路（第六版）	Electric Circuits, 6e	James W. Nilsson	路而红	中文版 英文版
	工程电路分析（第六版）	Engineering Circuit Analysis, 6e	William H. Hayt, Jr.	王大鹏	中文版 英文版
	射频电路设计——理论与应用	RF Circuit Design: Theory and Applications	Reinhold Ludwig	王子宇	中文版
	电子学原理	Eletronic Concepts: An Introduction	Jerrold H. Krenz	马爱文	中文版
	数字电路简明教程	Digital Electronics: A Simplified Approach	Robert D. Thompson	高　鹏	中文版
	逻辑电路设计基础	Introduction to Logic Design	Alan B. Marcovitz	殷洪玺	中文版
	数字集成电路设计	Digital Integrated Circuit Design	Ken Martin		英文版
	线性电路分析基础（第二版）	Elementary Linear Circuit Analysis	Leonard S. Bobrow		英文版
	半导体器件电子学	Semiconductor-Device Electronics	R. M. Warner		中文版 英文版
	电路原理（第七版）	Principles of Electric Circuits Conventional Current Version, 7e	Thomas L. Floyd	罗伟雄	中文版
信号与系统	信号与系统（第二版）	Signals and Systems, 2e	Alan V. Oppenheim		英文版
	信号与系统（第二版）	Signals and Systems, 2e	Simon Haykin	林秩盛	中文版 英文版
	应用Web和MATLAB的信号与系统基础（第二版）	Fundamentals of Signals and Systems Using the Web and MATLAB, 2e	Edward W. Kamen	高　强	中文版
	统计与自适应信号处理	Statistical and Adaptive Signal Processing	Dimitris G. Manolakis	周　正	中文版
	统计信号处理基础——估计与检测理论	Fundamentals of Statistical Signal Processing	Steven M. Kay	罗鹏飞	中文版
	自适应滤波器原理（第四版）	Adaptive Filter Theory, 4e	Simon Haykin	郑宝玉	中文版 英文版
数字信号处理	数字图像处理（第二版）	Digital Image Processing, 2e	Rafael C. Gonzalez	阮秋琦	中文版 英文版
	数字信号处理基础	Fundamentals of Digital Signal Processing	Joyce Van de Vegte	侯正信	中文版
	实时信号处理——信号处理系统的设计与实现	Real-Time Signal Processing: Design and Implementation of Signal Processing Systems	John G. Ackenhusen	彭启琮	中文版
	小波与傅里叶分析基础	A First Course in Wavelets with Fourier Analysis	Albert Boggess		英文版
	信号处理滤波器设计——基于MATLAB和Mathematica的设计方法	Filter Design for Signal Processing Using MATLAB and Mathematica	Miroslav D. Lutovac	朱义胜	中文版 英文版
	数字信号处理实用方法（第二版）	Digital Signal Processing: A Practical Approach, 2e	Emmanuel Ifeachor	罗鹏飞	中文版 英文版
	数字信号处理——频谱计算与滤波器设计	Digital Signal Processing: Spectral Computation and Filter Design	Chi-Tsong Chen		英文版
	数字信号处理——系统分析与设计	Digital Signal Processing: System Analysis and Design	Paulo S. R. Diniz	全子一	中文版 英文版
	LabVIEW 6i 实用教程	LabVIEW 6i Student Edition	Robert H. Bishop	朱世华	中文版
	VHDL 数字系统设计	Digital System Design with VHDL	Mark Zwolinski		英文版
	MATLAB 原理与工程应用	An Engineer's Guide to MATLAB	Edward B. Magrab	高会生	中文版

● 教材服务与教辅支持，请联系：010-68270519

国外电子与通信教材系列 ②

类别	中文书名	英文书名	作者	译审者	版别
微电子	半导体器件原理	Semiconductor device Fundamentals	Robert F. Pierret		中文版
	超大规模硅集成电路工艺技术——理论、实践与模型	Silicon VLSI Technology: Fundamental, Practice and Modeling	James D. Plummer	许 军	中文版 英文版
	低压低功耗CMOS/BiCMOS超大规模集成电路	CMOS/BiCMOS ULSI Low Voltage, Low Power	Kiat-Seng Yeo	周元兴	中文版
	CMOS射频集成电路设计	The Design of CMOS Radio-Frequency Integrated Circuits	Thomas H. Lee	周润德	中文版 英文版
	专用集成电路	Application-Specific Integrated Circuits	Michael J. S. Smith	章倩苓	中文版 英文版
	CMOS模拟电路设计（第二版）	CMOS Analog Circuit Design, 2e	Phillip E. Allen		中文版 英文版
	微电子制造科学原理与工程技术（第二版）	The Science and Engineering of Microelectronic Fabrication, 2e	Stephen A. Campbell	周润德	中文版 英文版
	超大规模集成电路与系统导论	Introduction to VLSI Circuits and Systems	John P. Uyemura	周润德	中文版
通信系统	通信系统（第四版）	Communication Systems, 4e	Simon Haykin	沈连丰	中文版 英文版
	通信系统工程（第二版）	Communication Systems Engineering, 2e	John G. Proakis	沈连丰	中文版
	电子通信系统（第四版）	Electronic Communications Systems: Fundamentals through Advanced, 4e	Wayne Tomasi	薛荣华	中文版 英文版
	数字与模拟通信系统（第六版）	Digital and Analog Communication Systems, 6e	Leon W. Couch, II	罗新民	中文版
	数字通信——基础与应用（第二版）	Digital Communications: Fundamentals and Applications, 2e	Bernard Sklar	沈连丰	中文版 英文版
	光纤通信（第三版）	Optical Fiber Communications, 3e	Gerd Keiser	李玉权	中文版
	非线性光纤光学原理及应用	Nonlinear Fiber Optics, Third Edition & Application of Nonlinear Fiber Optics	Govind P. Agrawal	贾东方	中文版
	现代通信光电子学（第五版）	Optical Electronics in Modern Communications, 5e	Amnon Yariv	陈鹤鸣	中文版 英文版
	CDMA蜂窝移动通信与网络安全	CDMA Cellular Mobile Communications and Network Security	Man Young Rhee	袁超伟	中文版 英文版
	宽带无线数字通信	Wideband Wireless Digital Communications	Andreas Molisch	姚 彦	中文版 英文版
	第三代移动通信系统原理与工程设计——IS-95 CDMA和cdma 2000	IS-95 CDMA and cdma2000: Cellular/ PCS Systems Implementation	Vijay K. Garg		英文版
	视频处理与通信	Video Processing and Communications	Yao Wang	侯正信	中文版
	无线通信调制与编码	Modulation and Coding for Wireless Communications	Alister Burr		英文版
电磁场与微波	天线（第三版）（上下册）	Antennas: For All Applications, 3e	John D. Kraus	章文勋	中文版
	射频与微波电子学	Radio Frequency and Microwave Electronics Illustrated	Matthew M. Radmanesh		英文版
	现代无线通信系统电波传播	Radio Propagation for Modern Wireless Systems	Henry L. Bertoni	徐承和	中文版 英文版
	电磁波理论	Electromagnatic Wave Theory	Jin Au Kong	吴 季	中文版

● 教材服务与教辅支持，请联系：010-68270519

目　　录

第1章　VLSI概论 ………… 1
1.1　复杂性与设计 ………… 1
1.1.1　设计流程举例 ………… 3
1.1.2　VLSI芯片的类型 ………… 5
1.2　基本概念 ………… 5
1.3　本书安排 ………… 7
1.4　参考资料 ………… 8

第1部分　硅片逻辑

第2章　MOSFET逻辑设计 ………… 13
2.1　理想开关与布尔运算 ………… 13
2.2　MOSFET开关 ………… 17
2.3　基本的CMOS逻辑门 ………… 23
2.3.1　非门(NOT门) ………… 24
2.3.2　CMOS或非门(NOR门) ………… 25
2.3.3　CMOS与非门(NAND门) ………… 28
2.4　CMOS复合逻辑门 ………… 30
2.4.1　结构化逻辑设计 ………… 32
2.4.2　异或门(XOR)和异或非门(XNOR) ………… 40
2.4.3　一般化的AOI和OAI逻辑门 ………… 41
2.5　传输门(TG)电路 ………… 42
逻辑设计 ………… 43
2.6　时钟控制和数据流控制 ………… 45
2.7　参考资料 ………… 48
2.8　习题 ………… 48
第3章　CMOS集成电路的物理结构 ………… 51
3.1　集成电路工艺层 ………… 51
互连线的电阻和电容 ………… 53
3.2　MOSFET ………… 56
3.2.1　硅的导电性 ………… 58
3.2.2　nFET和pFET ………… 62
3.2.3　FET中的电流 ………… 63
3.2.4　栅电容的驱动 ………… 67

3.3 CMOS 工艺层 …… 69
3.4 FET 阵列设计 …… 71
3.4.1 基本门设计 …… 73
3.4.2 复合逻辑门 …… 76
3.4.3 一般性讨论 …… 78
3.4.4 小结 …… 81
3.5 参考资料 …… 81
3.6 习题 …… 82
第 4 章 CMOS 集成电路的制造 …… 85
4.1 硅工艺概述 …… 85
本章概要 …… 87
4.2 材料生长与淀积 …… 88
4.2.1 二氧化硅 …… 88
4.2.2 氮化硅 …… 89
4.2.3 多晶硅 …… 90
4.2.4 金属化 …… 90
4.2.5 掺杂硅层 …… 91
4.2.6 化学机械抛光 …… 92
4.3 刻蚀 …… 93
洁净间 …… 97
4.4 CMOS 工艺流程 …… 97
工艺改进 …… 101
4.5 设计规则 …… 104
4.5.1 物理极限 …… 107
4.5.2 电气规则 …… 108
4.6 参考资料 …… 108
第 5 章 物理设计的基本要素 …… 109
5.1 基本概念 …… 109
CAD 工具 …… 110
5.2 基本结构的版图 …… 111
5.2.1 n 阱 …… 111
5.2.2 有源区 …… 112
5.2.3 掺杂硅区 …… 113
5.2.4 MOSFET …… 114
5.2.5 有源区接触 …… 117
5.2.6 金属层 1 …… 118
5.2.7 通孔和多层金属 …… 121
5.2.8 防止闩锁现象 …… 121
5.2.9 版图编辑器 …… 123
5.3 单元概念 …… 124
5.4 FET 的尺寸确定和单位晶体管 …… 129

5.5 逻辑门的物理设计 …… 133
5.5.1 NOT 单元 …… 134
5.5.2 与非门(NAND)和或非门(NOR)单元 …… 135
5.5.3 复合逻辑门 …… 136
5.5.4 关于版图的小结 …… 136
5.6 设计层次化 …… 137
5.7 参考资料 …… 139

第 2 部分　从逻辑到电子电路

第 6 章　MOSFET 的电气特性 …… 143
6.1 MOS 物理学 …… 143
阈值电压的推导 …… 146
6.2 nFET 电流-电压方程 …… 147
6.2.1 SPICE Level 1 方程 …… 153
6.2.2 体偏置效应 …… 153
6.2.3 电流方程推导 …… 154
6.3 FET 的 RC 模型 …… 157
6.3.1 漏源 FET 电阻 …… 157
6.3.2 FET 电容 …… 159
6.3.3 模型建立 …… 163
6.4 pFET 特性 …… 165
pFET 寄生参数 …… 168
6.5 小尺寸 MOSFET 模型 …… 169
6.5.1 尺寸缩小原理 …… 169
6.5.2 小尺寸器件效应 …… 171
6.5.3 SPICE 模型 …… 172
6.6 参考资料 …… 173
6.7 习题 …… 174
第 7 章　CMOS 逻辑门电子学分析 …… 176
7.1 CMOS 反相器的直流特性 …… 176
7.2 反相器的开关特性 …… 181
7.2.1 下降时间计算 …… 185
7.2.2 上升时间 …… 186
7.2.3 传播延时 …… 188
7.2.4 一般分析 …… 189
7.2.5 反相器电路小结 …… 191
7.3 功耗 …… 191
7.4 DC 特性:与非门(NAND 门)和或非门(NOR 门) …… 193
7.4.1 与非门(NAND 门)分析 …… 193
7.4.2 或非门(NOR 门) …… 196
7.5 与非门和或非门的暂态响应 …… 197

7.5.1 NAND2 开关时间 …… 198
7.5.2 二输入或非门(NOR2)的开关时间 …… 200
7.5.3 小结 …… 202
7.6 复合逻辑门的分析 …… 202
功耗 …… 204
7.7 逻辑门过渡特性设计 …… 205
7.8 传输门和传输管 …… 208
7.9 关于 SPICE 模拟 …… 211
7.10 参考资料 …… 213
7.11 习题 …… 214
第 8 章 高速 CMOS 逻辑电路设计 …… 217
8.1 门延时 …… 217
8.2 驱动大电容负载 …… 224
在反相器链中使延时最小 …… 226
8.3 逻辑努力(Logical Effort) …… 231
8.3.1 基本定义 …… 231
8.3.2 一般化情形 …… 235
8.3.3 级数的优化 …… 239
8.3.4 逻辑面积 …… 240
8.3.5 分支情况 …… 241
8.3.6 小结 …… 242
8.4 BiCMOS 驱动器 …… 242
8.4.1 双极型管的特性 …… 243
8.4.2 驱动电路 …… 246
8.5 参考资料 …… 248
8.6 习题 …… 249
第 9 章 CMOS 逻辑电路的高级技术 …… 251
9.1 镜像电路 …… 251
9.2 准 nMOS 电路 …… 253
9.3 三态电路 …… 255
9.4 时钟控制 CMOS …… 256
9.5 动态 CMOS 逻辑电路 …… 261
9.5.1 多米诺逻辑 …… 263
9.5.2 动态逻辑电路的功耗 …… 266
9.6 双轨逻辑电路 …… 266
9.6.1 CVSL …… 267
9.6.2 互补传输管逻辑 …… 269
9.7 参考资料 …… 270
9.8 习题 …… 270

第3部分 VLSI 系统设计

第10章 用 Verilog® 硬件描述语言描述系统 …… 275

10.1 基本概念 …… 275

10.2 结构化的门级模型 …… 276

Verilog 举例 …… 276

10.3 开关级建模 …… 284

10.4 层次化设计 …… 288

10.5 行为级和 RTL 建模 …… 291

10.6 参考资料 …… 297

10.7 习题 …… 298

第11章 常用的 VLSI 系统部件 …… 300

11.1 多路选择器 …… 300

11.2 二进制译码器 …… 305

11.3 相等检测器和比较器 …… 307

11.4 优先权编码器 …… 310

11.5 移位和循环操作 …… 313

11.6 锁存器 …… 316

11.7 D 触发器 …… 321

11.8 寄存器 …… 326

11.9 综合的作用 …… 327

11.10 参考资料 …… 328

11.11 习题 …… 329

第12章 CMOS VLSI 运算电路 …… 330

12.1 一位加法器电路 …… 330

12.2 串行进位加法器 …… 335

12.3 超前进位加法器 …… 338

12.3.1 曼彻斯特进位链 …… 343

12.3.2 扩展为宽位加法器 …… 345

12.4 其他高速加法器 …… 348

12.4.1 进位旁路电路(Carry-Skip Circuits) …… 348

12.4.2 进位选择加法器(Carry-Select Adder) …… 349

12.4.3 进位保留加法器(Carry-Save Adder) …… 350

12.5 乘法器 …… 351

12.5.1 阵列乘法器 …… 353

12.5.2 其他乘法器 …… 355

12.6 小结 …… 358

12.7 参考资料 …… 358

12.8 习题 …… 359

第 13 章 存储器与可编程逻辑 …… 360
13.1 静态 RAM …… 360
13.2 SRAM 阵列 …… 364
13.3 动态 RAM …… 372
13.3.1 DRAM 单元的物理设计 …… 375
13.3.2 分割字线结构 …… 376
13.4 ROM 阵列 …… 377
用户编程 ROM …… 379
13.5 逻辑阵列…… 382
13.5.1 可编程逻辑阵列 …… 382
13.5.2 门阵列…… 385
13.6 参考资料…… 386
13.7 习题…… 387
第 14 章 系统级物理设计 …… 390
14.1 大规模集成电路的物理设计…… 390
14.2 互连线延时模型…… 391
14.2.1 信号延时与连线长度的关系 …… 398
14.2.2 对互连线延时的考虑 …… 399
14.3 串扰…… 399
有关串扰的考虑 …… 403
14.4 互连线的尺寸缩小…… 404
14.5 布局布线…… 406
14.6 输入和输出电路…… 410
14.6.1 输入电路…… 410
14.6.2 输出驱动器 …… 414
14.7 电源的分配和功耗…… 416
同时切换引起的噪声 …… 418
14.8 低功耗设计考虑…… 421
14.9 参考资料…… 422
14.10 习题 …… 423
第 15 章 VLSI 时钟和系统设计 …… 426
15.1 时钟控制触发器…… 426
经典的状态机 …… 427
15.2 CMOS 时钟方式 …… 429
15.2.1 钟控逻辑链 …… 429
15.2.2 动态逻辑链 …… 437
15.3 流水线系统…… 439
15.4 时钟的产生和分配…… 443
15.4.1 时钟的稳定和产生 …… 444
15.4.2 时钟布线与驱动器树结构 …… 446

15.5 系统设计考虑…………………………………………………………………… 451
15.5.1 位片式设计 …………………………………………………………… 452
15.5.2 cache 存储器 ………………………………………………………… 453
15.5.3 脉动系统和并行处理 ………………………………………………… 454
15.5.4 小结 …………………………………………………………………… 455
15.6 参考资料………………………………………………………………………… 455
第 16 章 VLSI 电路的可靠性与测试 ………………………………………………… 457
16.1 一般概念………………………………………………………………………… 457
可靠性建模 ………………………………………………………………………… 459
16.2 CMOS 测试 ……………………………………………………………………… 462
16.2.1 CMOS 故障模型 ……………………………………………………… 463
16.2.2 门级测试………………………………………………………………… 464
16.2.3 I_{DDQ}测试 ………………………………………………………… 466
16.3 测试生成方法…………………………………………………………………… 467
16.3.1 静态 CMOS 逻辑门 …………………………………………………… 467
16.3.2 故障的逻辑影响 ……………………………………………………… 468
16.3.3 路径的敏化 …………………………………………………………… 470
16.3.4 D算法…………………………………………………………………… 470
16.3.5 布尔差分 ……………………………………………………………… 471
16.4 小结……………………………………………………………………………… 473
16.5 参考资料………………………………………………………………………… 473

第1章　VLSI概论

VLSI是**大规模集成电路**的英文缩写。这个不那么一目了然的学术名称集中反映分析及设计密集型集成电路所涉及的电子工程和计算机工程的许多领域。虽然很难给出一个严格的定义，但通常划分的标准是指一个VLSI含有约100万(10^6)个以上的开关器件或逻辑门。21世纪初，在一个较为复杂的设计中，一个典型的约1 cm见方的**芯片**上**晶体管**(开关器件)的实际数目已经超过1亿个。

本书介绍**数字VLSI芯片设计**的基础。重点介绍如何将一个系统的设计要求转移到一个小硅片上。这一过程有许多技术上的细节。其中有一些说明和分析显而易见，而另一些则可能要到后面几章才能理解。之所以这样是因为VLSI工程包含几个不同的"专门领域"，它们以一种独有的方式交织在一起。学习VLSI的最困难之处在于理解将不同领域联系在一起的共同目的，一旦掌握了这一点，就能以自己的方式理解这个当代最具魅力的领域之一。

1.1　复杂性与设计

工程上实现一个VLSI芯片是一项极为复杂的任务。图1.1所示的"VLSI设计漏斗"概念可以帮助外行们易于理解这一领域。它显示这样一个过程：我们提供基本的需要，例如资金，构想及市场信息等，然后将它们统统倒入一个"技术魔法漏斗"，再加入一堆沙子作为原材料，漏斗底部就会生产出超级芯片。它将被销售出数百万片，且有可能变革整个世界，还会使某人致富。当然，这一过程需要工程师和科学家们参加，不过他们只是把那些东西放到一起。可惜的是这个过程比起图中所画的要稍微复杂一些。

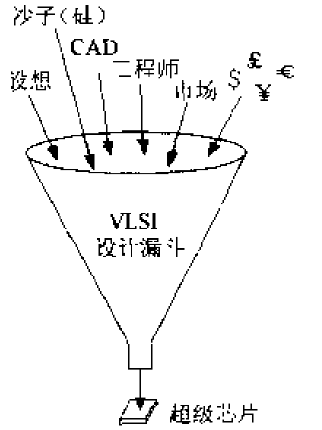


图1.1　VLSI设计漏斗

任何一个含有上百万个元件的系统本来就是很难理解的。一个人的大脑不可能处理设计和实现所需要的如此复杂的信息。组建一支**设计队伍**是完成一个VLSI项目的现实方法，因为这可以使每个人只需研究整个系统的一小部分。在一个现代设计项目中，可能会有数百名工程师和科学家从事设计中不同部分的工作。由于整个队伍是在完成同一个项目，因此对于它的每个成员来说，多少了解一下自己的工作在总目标中所处的位置是很重要的。这是通过**设计层次化**来实现的；即芯片被看做有许多不同的"层次"，从抽象到物理实现。每个层次都是重要的，每个层次又有数个分支，每个分支都可能逐渐成为一个人的终身事业。

在介绍VLSI时，我们将不断地强调这样一个事实，即这个领域在本质上生来就是一个多

学科的领域,需要无数方面的专家共同实现一个能够正确工作的设计。计算机结构设计人员必须与程序编写人员及逻辑设计人员交流合作,他们还必须理解电路设计及硅片加工的某些问题。电子专业人员必须超出电路的范围,看看他们研究的元件如何影响系统。每个人都需要依靠计算机辅助设计工具及支持小组,支持小组完成 10 000 个左右的其他任务,在此就不赘述了。如果这种说法使这个领域听起来十分复杂,那是因为它就是如此。VLSI 不是一个容易理解的简单学科,但是经过一段适当的时间是能够掌握它的基础的。最终在这个领域工作的人往往会被吸引住,因为其中的一个或多个方面会引起他们的兴趣,并且符合他们的知识背景。

既然对 VLSI 所涉及的范围有所了解,就让我们进一步来描述它的设计过程。图 1.2 显示它的概貌及在设计过程中的各个主要步骤。VLSI 设计的起始点是对系统要求的说明。在这一点上,产品同时用一般的和专门的术语定义,即提出整个项目的设计目标,如功能、速度、尺寸等等。这是层次化设计的最高层。系统说明用来建立抽象的高层次的模型。数字设计通常基于某种**硬件描述语言**(HDL),它可以建立操作的抽象模型。VHDL 和 Verilog™是实际中最常用的硬件描述语言,其他几种语言(包括 C 和 C++)也被应用。抽象模型包含系统中每个模块的功能及各个模块间相互作用的信息。这个模型必须经过广泛的验证步骤,一遍遍地检查设计以确保正确无误。

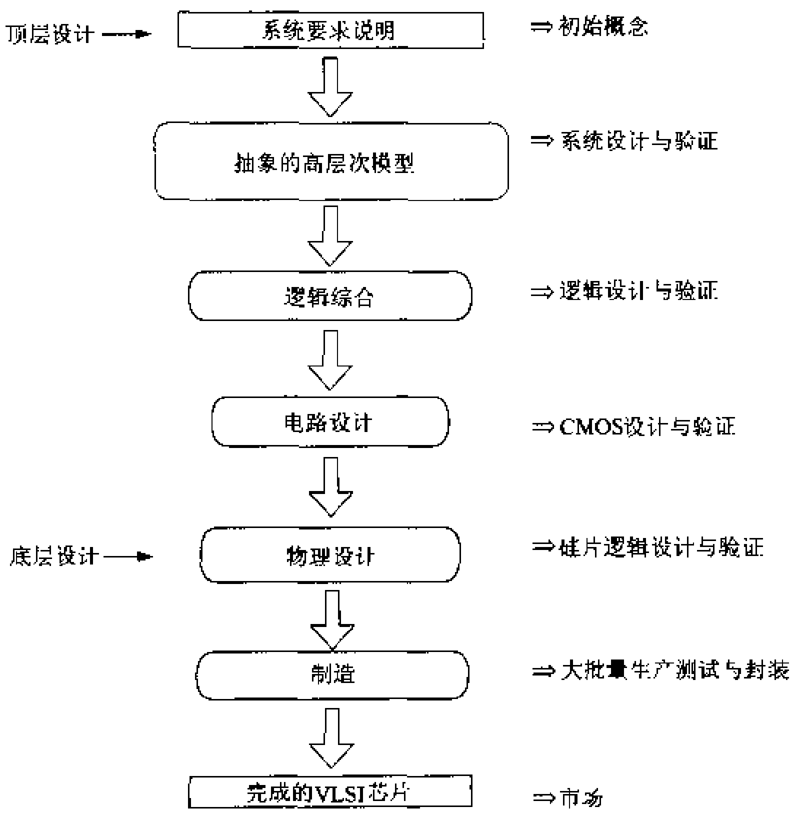


图 1.2 层次化设计的一般框图

设计过程的下一个步骤叫做**综合**。通过抽象的逻辑模型,完成系统的**逻辑设计**,即说明构成每个单元所需要的基本门和基本单元。这是将设计转入**电路设计层**的基础。这里采用晶体管作为开关,而布尔变量则看成变化的电压信号。为了构成晶体管,我们再向下一层转入**物理**

设计。在这一层,采用复杂的映射技术将电路建立在一个微小的硅片上,晶体管和导线转变成用金属和其他材料做成的极其微小的形式。物理设计是系统设计的最后一步。设计完成后,项目进入生产线,其最终结果是一个完成的电子 VLSI 芯片。

如果从系统一级的要求出发,整个设计过程叫做**自上而下**的方法。起始的工作是相当抽象和理论性的。在许多步骤完成之前,这些工作与硅片没有直接的联系。反过来的方法起始于硅片和电路阶段。第一步是建立最基本的单元,如逻辑门、加法器、寄存器等。这些单元组合起来就可得到较大、较复杂的逻辑模块,后者又可用做更大规模设计的功能块。这一**自下而上**的设计方法用于小项目尚可,对复杂的现代 VLSI 设计就不切实际了。如果从一位一位数开始设计一个工作正确的 16 位微处理机,那将是极端困难的。

自下而上地学习 VLSI 的各个方面则是学习这一领域基础的好办法,因此本书的前半部分即采用这一方法。我们将从简单开始,逐渐进入复杂、抽象的更高层次。目的是给大家一个连贯的认识,即这个领域是一个由许多不同领域组成的整体。有些问题即便看上去极度专业化,以后也会与其他概念相关联。一旦掌握了基础,我们就可以站在更高的层次上研究问题。本书的后半部分将介绍 VLSI 系统方面的问题,以给出一个完整的概念。

1.1.1 设计流程举例

作为一个层次化设计的例子,让我们先确定在一个基本的微处理器设计中什么是必需的。初始的构想可在定义指令集和部件的系统层次上。一条指令是所设计的微处理器能够执行的一个基本操作(如两个二进制数相加)。指令集则是一个具体处理器的全部指令集合。部件是提供某个指定功能(如加法)的数字逻辑单元。计算机体系结构领域则研究组成计算机的单元及它们是如何组合在一起的。

图 1.3 为这一问题的基本设计流程。指令集和部件组可以来建造一个高层次的结构模型。这个层次只是抽象地描述系统的行为而不考虑在实际构成电路时需要的低层次细节。例如,我们可以定义一个加法运算为

$$\text{register_X} \leftarrow A + B$$

意即将 A 和 B 的和传送到一个名为 register_X(寄存器 X)的存储器件中。这种高层次的抽象可以用来定义处理器的体系结构,且常称为寄存器传输级(RTL)描述。RTL 模型只描述系统如何操作,不考虑具体的部件。当用 HDL 语言把它写下后,就可用来测试指令并且验证体系结构的行为。抽象设计能为系统建立一个方框图。

RTL 代码可以转化为一个包含更多有关操作和行为部件细节的等效描述。每个功能块的操作可在 HDL **行为级**上进行概括,这时的重点是放在这些功能块与其他部分相互作用的大行为上。这一阶段的行为建模是极为关键的,因为要用它来验证体系结构。这一阶段的所有问题都必须得到解决后才能进行下一步工作。

下一阶段的设计过程是将系统的功能块转化为以布尔方程和逻辑门为基础的逻辑模型。这使抽象设计进入实际阶段,它是实现硬件设计的第一步。这一阶段可以通过两种途径完成:逻辑电路的**自动设计与综合**或**定制设计**。自动设计以一系列在高性能工作站运行的 CAD(计算机辅助设计)工具为基础。一个综合工具通常可以接受 HDL 代码,按照一组预先确定的规则产生相应的逻辑电路。编写恰当的 HDL 代码能够很快地产生逻辑设计,所以自动综合用于

所有非关键部分。当特殊问题出现或综合的结果不能满足必需的要求时，就要采用定制设计，进行各种逻辑方程和电路的推导和测试，作为解决手头问题的手段。这是一项高强度和耗时的工作，只用于关键部分。

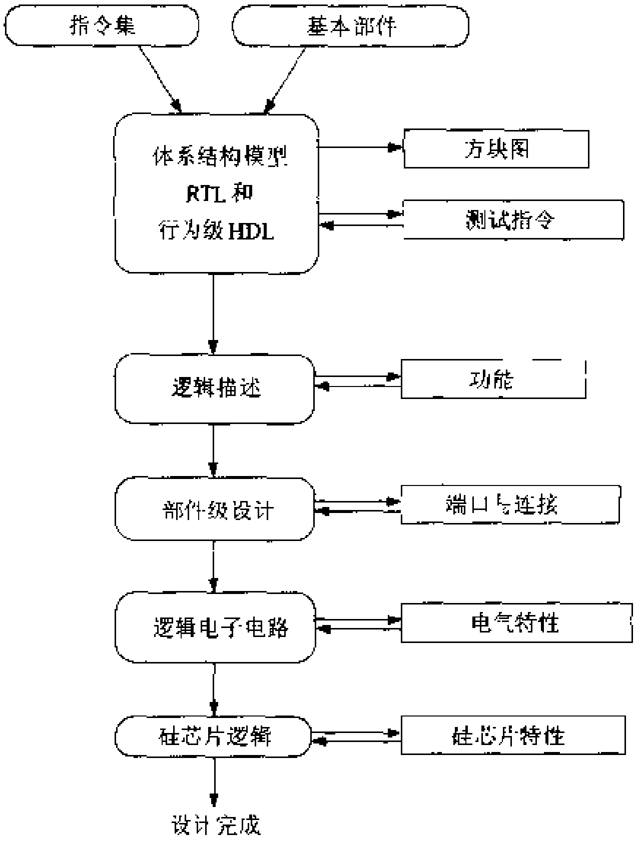


图 1.3 一个微处理器的简单设计流程

逻辑模型产生功能部件，部件再转化为电子电路。在这一设计过程中，硅片电路的特性显得重要起来。对于一个复杂的功能，通常可以找到几个等效的逻辑表达式，它们给出相同的输出，却使用不同的方程和逻辑门。比如，回想一下如何用一张卡诺图来简化逻辑方程的。硅片 VLSI 之所以复杂是因为它的每类逻辑门或电路都具有各自不同的特性，必须经常寻找比用一个明显可行的方案设计更快或更小的电路。一个先进的综合工具能够接受 HDL 代码且对逻辑门和硅片电路提出设计建议，但是从任何意义上来讲，这套工具还未能达到产生最佳设计的水平。

逻辑和电路设计完毕后，是用这些信息在物理设计层次上产生集成电路。这个过程需要经过一系列步骤完成。此时晶体管定义为在硅芯片上的三维结构，然后利用另一组 CAD 绘图工具布局和布线。一旦完成，可对设计进行测试和验证，用来生成生产线实际制造电子芯片所需要的数据库。制造 VLSI 芯片本身是一个非常复杂的专业领域，一旦开始制造，可能要用好几个星期才能生产出最后的电路。

整个设计过程的细节远比简单流程图所画的要复杂得多，不过它还是说明了自上而下设计流程最基本的内容。VLSI 设计应考虑必要的细节使制造出的芯片能按设计要求高度可靠地工作并具有较长的寿命，当然销售还要赢利。

1.1.2　VLSI 芯片的类型

就工程上说，数字 VLSI 芯片可以按其实现和构成电路时所采用的方法分类。**全定制**设计是指一个设计项目的每个电路都是定制设计的。这是一个非常单调耗时的过程，因此用这种方法设计整个系统不切实际。

专用集成电路(ASIC)使数字电路设计者产生一个用于专门用途的集成电路(IC)，它普遍用于设计原型和小批量产品。它们使用一套扩展的 CAD 工具。这些工具通过标准的数字逻辑结构，如状态图、功能表和逻辑图来描述系统。通常，一个 ASIC 设计者并不需要了解有关电子器件或硅芯片的实际结构。设计自动化 CAD 工具负责进行逻辑设计并且构成芯片的大部分。ASIC 设计的一个缺点是，所有的特性(如速度)都由体系结构设计确定，设计者无法改动电子器件，所以时间延迟不能改变。现代 ASIC 已发展到很高的技术水平，一般都能解决一大类问题。

半定制设计介于全定制和 ASIC 类电路的设计之间。芯片的大部分采用一组预先定义的基本单元，作为构建功能块。每个单元提供一个基本的功能，例如逻辑操作或存储电路，因此主要设计取决于一个叫做“库”的数据库集合。一个单元本身包含在硅片上建立电路所需的所有信息。如果采用单元库还不能满足系统要求，那么半定制设计可以允许设计者找到一种解决办法，产生具有所需要特性的另一种硅片电路。这仅用于芯片上有问题的小范围。例如微处理器中的浮点电路可以十分复杂，所以某些部分需要运用定制设计来满足时序方面的安排。大多数高性能的芯片都要用到各种半定制的设计方法。

1.2　基本概念

本书的目的是介绍 VLSI 领域的全貌。总的来说，VLSI 设计是一门系统设计学科。其中许多方面的讲解并不需要硅片电路的基础。利用 CAD 工具可以完成系统的设计，且把必要的数据交给生产部门用于生产。虽然这种方法可以得到功能正确的结果，但却使其中许多细节不为设计者所见。简化设计程序是重要的，但许多 VLSI 功能最强的技术和概念处于较低层次，因面会被丢失。电路虽能运行，但不能像它们本来可以的那样快或那样小。

应当把 VLSI 看做一门关于复杂集成电路的概念、设计和制造的单一学科。许多系统层次上的概念都是以硅片层次上的电路特性为基础的。当 Caltech 公司的 Carver Mead 在 20 世纪 70 年代开拓这一领域时，VLSI 的最重要基础之一来源于他观察到可以把数字电子集成电路看成在硅芯片表面上的一组几何图形。图形组合代表不同的逻辑功能，且在系统中重复多次，因而复杂问题可以通过一定的结构方式和重复的图形恰当地放在一起的概念进行处理。信号流和数据的移动可以通过追踪载流金属线的路径来得到。这样，就有可能将布尔表达式以一种确切定义的方式直接转化为硅片上的几何图形。图 1.4 中 CMOS 芯片的显微照片显示在一个已完成的器件上许多这方面的特点。特别注意那些重复的图形，顺序放置的矩形线条，多边形以及几何图形的组合。[①] Mead 的观察(及其大量的工作)已将 VLSI 构建成今天这

① 这是 Georgjia Tech 设计的一个二进制加法器电路的一部分，每组图形相加两位并产生和位及进位输出。

样一个重要的领域。可见，对 VLSI 有一个全面一致的认识已变得非常重要。

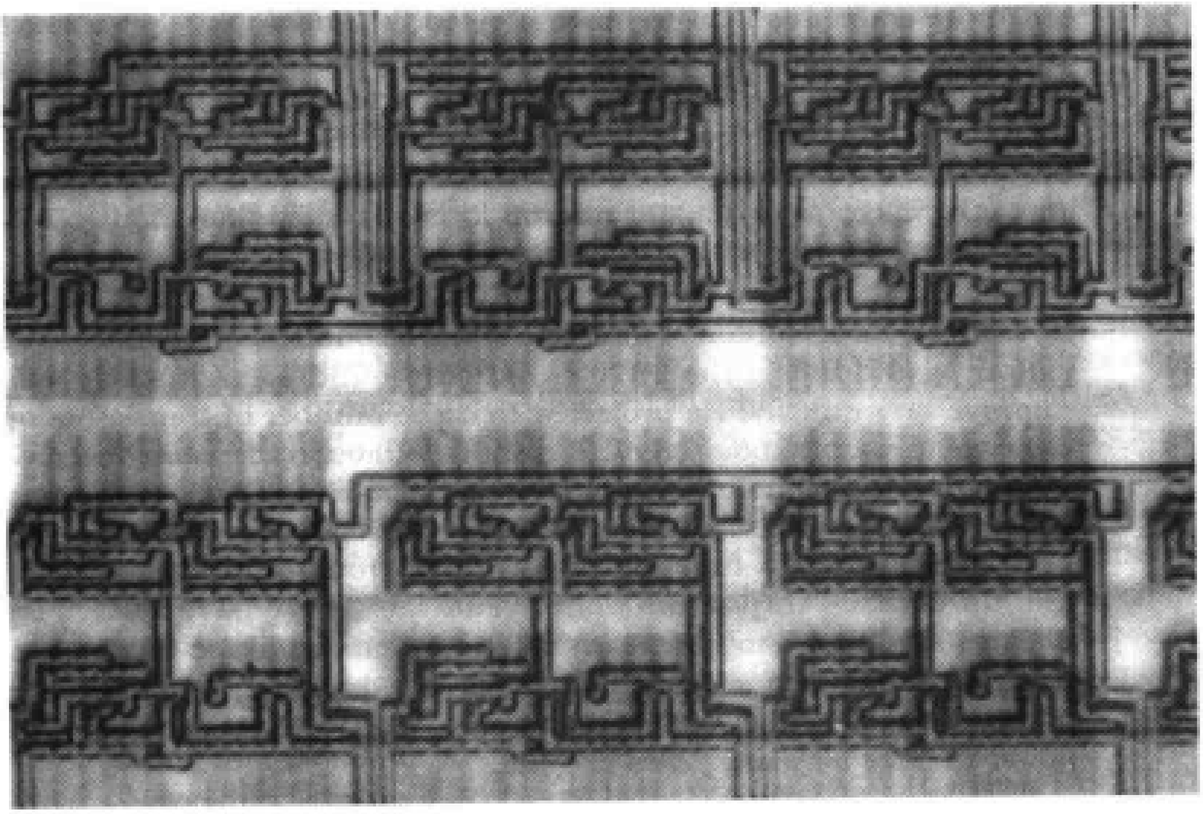


图 1.4 一个数字 CMOS 集成电路局部的显微照片

VLSI 设计包含数字系统设计中许多实际的方面。一个事实是，即便性能最强的**系统芯片**(SOC)也必须与其他部件发生联系才能得到可工作的单元。这是通过将硅片电路放在一片矩形材料的中央，然后以某种技术使外部导线与其相连。图 1.5 为**压焊块**的应用，它们是一些方形的金属块，导线可以被焊接并且连接到安装芯片的封装上。IBM 开发一种称为 C4 的更先进的技术，它能使金属"焊点"分布在芯片表面上，通过将芯片倒装使焊点朝下与布线栅格对准。

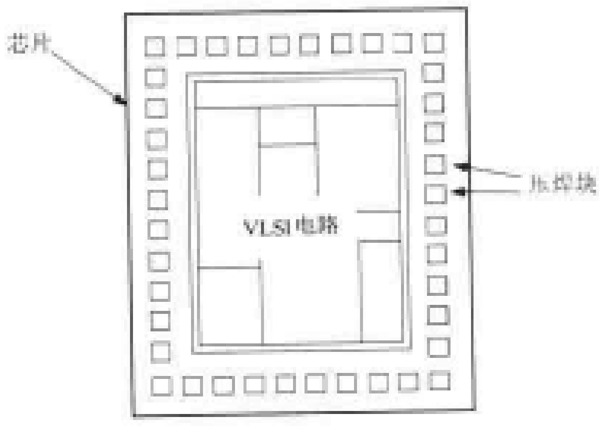


图 1.5 对外连接用的压焊块结构

在理想情况下，可将芯片做成任意大小，这似乎可以在一个芯片上设计越来越复杂的系统

而不需要任何焊接。可惜的是,由于硅晶体结构不可避免的缺陷,造成不能生产一个功能正确的设计。电路的面积越大,缺陷发生的可能性就越大,甚至仅一个坏的晶体管或连接点就会导致电路不工作。所以,尽可能使整个芯片的尺寸较小。我们还注意到,生产中的其他问题也限制了芯片的大小。

为了克服这一限制,我们借用这一理由:即缩小晶体管的尺寸就能把更多的器件放到一个给定的芯片面积上。从技术上讲,这是一个非常困难的问题。精细的设计和制造技术可使器件的最小尺寸达到 1.3×10^{-7} m 左右。达到这一水平时,可以改用微米(μm)即“**micron**”来度量。1 μm = 10^{-6} m = 10^{-4} cm。此时,称它为 0.13 μm 工艺。

对 VLSI 晶体管密度的经典预测之一称为**摩尔定律**(Moore's Law)。戈登·摩尔 (Gordon Moore)是 Intel 公司的创始人之一,他在 20 世纪 70 年代就看到了芯片制造技术将快速发展。他预计,在一个芯片上晶体管的数目大约每 18 个月就将翻倍。虽然由于技术上的问题或经济变缓而会有所不同,但摩尔定律已被证明与实际趋势惊人地接近。图 1.6 是随机从主要销售商处选取的微处理器芯片的器件数目随年度变化的关系图。由于尺寸的缩小受到技术上的限制,管子数目这样的增长速度还能保持多久,一直引起人们的争论。然而不管实际的增长率将如何,有一点是清楚的,即 VLSI 设计仍将在今后许多年中保持强劲的势头。

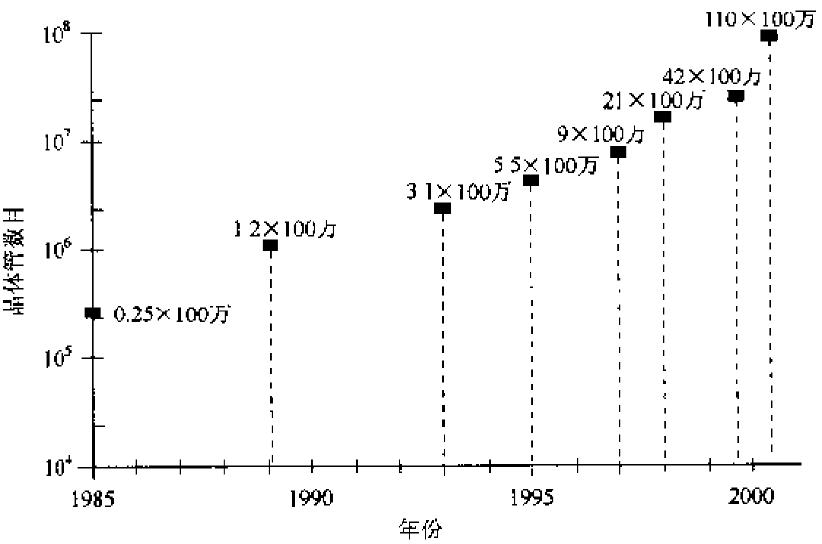


图 1.6 器件数目随年度增长

通过以上对 VLSI 某些问题的简短介绍,可以看到这一领域本身的广阔天地。一个 VLSI 设计小组的任务就是在一片细小硅片上制造一个复杂的大系统。从抽象建模和时序向下直到制作上百万个晶体管的芯片,设计小组在各个层次上都面临着各种约束。因此项目进展情况的介绍,工程概括以及严格的期限总是存在的。

欢迎来到令人振奋的 VLSI 世界!

1.3 本书安排

本书分为 3 个主要部分。对于自学者来说,虽然初次阅读时不一定每部分都看,但最好还

是按顺序学习。

第1部分为硅片逻辑，包括第1至第5章，内容为研究硅片逻辑电路的设计技术；主要集中在介绍晶体管逻辑电路，以及如何将它们转化为在硅芯片上的图案；介绍CMOS工艺流程的细节且把它们应用到实际芯片的设计中。在完成第1部分学习后，读者将能在电路及硅片层次上设计出无数个CMOS逻辑门。

第2部分为逻辑与电路的相互关系，它覆盖VLSI的电子电路部分，包括第6至第9章。其中第8章，第9章涵盖更深层的概念。该部分介绍晶体管的开关特性，并且用来分析数字电子逻辑门。内容涉及许多细节，但主要集中于影响系统性能和开关速度的重要基础知识。在完成这部分学习后，将能牢固理解逻辑设计与电气特性间的关系。

系统层次方面的问题放在第3部分，即VLSI系统设计，包括第10至第16章。该部分介绍Verilog® HDL的基础，作为系统层次上建模的工具。第11至第13章研究许多VLSI逻辑部件，如多路选择器、加法器及存储器等。第14和第15章介绍大规模芯片的设计问题。最后一章介绍数字测试作为本书的结束。

作者在完成本书时侧重于解释细节，努力使材料具有可读性和条理性。这一点在第1部分格外明显，读者在这里读到的许多内容一般不会在其他课程中找到。

所以，让我们不再迟缓，现在就开始进入这个超大规模集成电路的世界。

1.4 参考资料

[1] Dan Clein, **CMOS IC Layout**, Newnes Publishing Co., Boston, 2000.

[2] Randy H. Katz, **Contemporary Logic Design**, Benjamin-Cummings Publishing Co., Redwood City, CA, 1994.

[3] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2000.

[4] Jan Rabaey, **Digital Integrated Circuits**, Prentice-Hall, Upper Saddle River, NJ, 1996.

[5] Michael John Sebastian Smith, **Application-Specific Integrated Circuits**, Addison-Wesley Longman Inc., Reading, MA, 1997.

[6] John P. Uyemura, **A First Course in Digital Systems Design**, Brooks-Cole Publishers, Pacific Grove, CA, 2000.

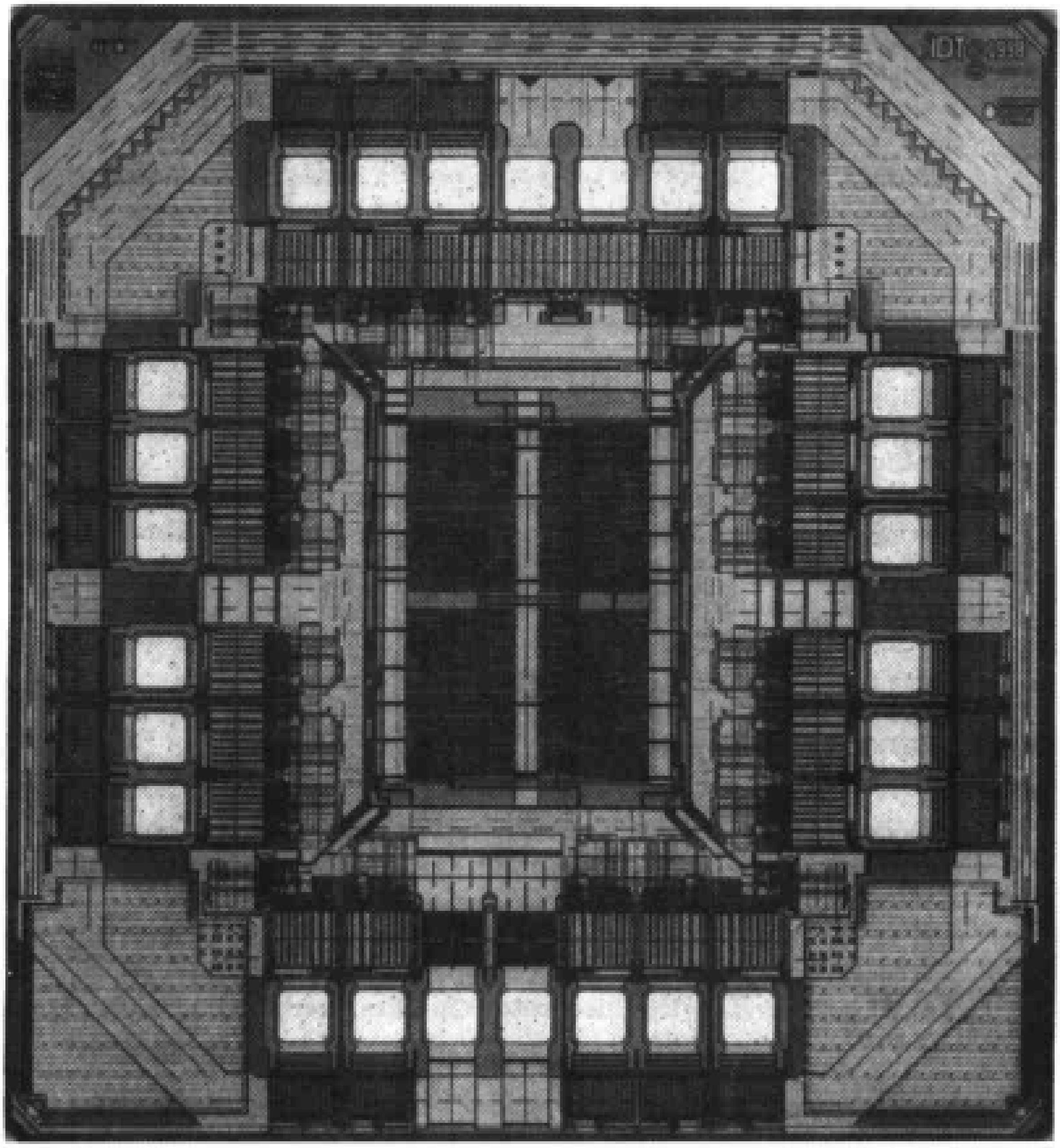
[7] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Press, Norwell, MA, 1999.

[8] John P. Uyemura, **Physical Design of CMOS Integrated Circuits Using L-Edit®**, PWS /Brooks-Cole Publishers, Pacific Grove, CA, 1995.

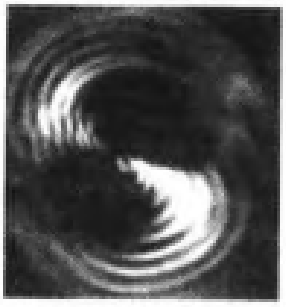
[9] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.

[10] Neil H.E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley Publishing Co., Reading, MA, 1993.

[11] Wayne Wolf, **Modern VLSI Design**, 2nd ed., Prentice-Hall PTR, Upper Saddle River, NJ, 1998.


IDT

第 1 部分



硅片逻辑

本页已使用福昕阅读器进行编辑。
福昕软件(C)2005-2010，版权所有，
仅供试用。

第 2 章　MOSFET 逻辑设计

CMOS 集成电路以名为 MOSFET 的双向器件作为逻辑开关。本章研究 MOSFET 的逻辑特性，介绍设计数字电路的技术。

2.1　理想开关与布尔运算

所有的数字设计都基于最基本的逻辑运算。我们在研究 VLSI 时的第一个任务就是要构建一个电子逻辑门，用它作为复杂开关电路的功能部件。

逻辑门是用一组**控制开关**构成的。一个**高电平有效**控制开关的特性可以通过图 2.1 来说明。在这一理想化的情况下，开关的状态（开或关）是由控制变量 A 的值来决定的。图 2.1 (a)中，控制位的值 $A=0$，定义为使开关断开，这意味着变量 x 和 y 之间没有联系，如图中左右两端被间断所代表的那样。相反的情况是开关闭合，如图 2.1(b)所示，开关的上部被"下推"。这种情况发生在 $A=1$ 时，此时开关两端相连，所以

$$y = x \tag{2.1}$$

成立。如果将左边的变量 x 定义为输入而右边的定义为输出，那么可以说，当 $A=1$ 时允许输入变量通过开关且建立输出值。之所以称为高电平控制开关，是因为需要一个 $A=1$ 的高电平控制位来闭合电路。

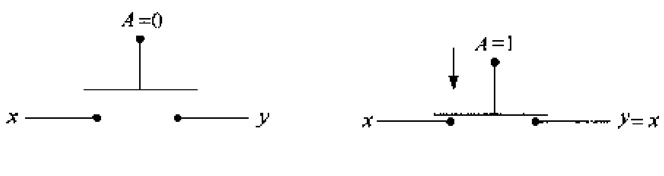


图 2.1　高电平有效开关的工作情况

这一开关的行为特征可以用另一种方式，即写出逻辑方程[①] 表示：

$$y = x \cdot A \quad \text{iff} \quad A = 1 \tag{2.2}$$

可以看到，当 $A=0$ 时，x 与 y 之间的关系是不确定的。虽然这看来是一个严重的缺点，但在实际中，可以增加一个开关来确定这种情况下 y 的值而避免这个问题。

让我们着手把理想开关的构想与电压源联合起来，构造一个逻辑电路。假设将两个由独立变量 a 和 b 控制的开关如图 2.2 所示连接起来，这两个开关称为相互**串联**。当跟踪信号通

① 我们用缩写数学符号"iff"表示"if and only if"（即"当且仅当……的时候"）。

过第一个开关时，由方程式(2.2)可知，这个开关的输出值为 $a\cdot 1$，如图 2.2 所示。这个值为第二个开关的输入，所以再次应用方程(2.2)得到最终的输出值：

$$g=(a\cdot 1)\cdot b=a\cdot b \tag{2.3}$$

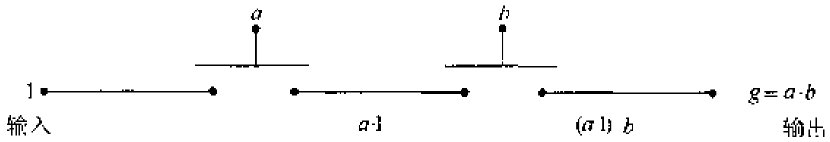


图 2.2 开关串联

这很容易用定性分析来说明：两个开关必须在 $a=1$ 与(AND) $b=1$ 时同时闭合才能使输入的 1 到达输出端得到 $g=1$。显然该电路提供了 AND2 的操作。[②] 但要注意，方程(2.2)仅在控制位的值为 1 时成立。如果为 0，那么开关左右两边就没有直接的关系。这两个输入还有另外三种可能性：

$$(a,b)=(1,0),(0,1),(0,0) \tag{2.4}$$

上面任何一种输入组合都应使逻辑输出 $g=0$，但从逻辑方程来看 g 却是不确定的。

在做进一步研究前，先弄清画逻辑电路的方法。通常开关图叫做**线路图**，因为它图示如何布线。我们将这一专有名词延用到含有电子器件的图中。为使所画的图比较紧凑清晰，画图中导线常会互相交叉。这时，将采用如图 2.3 所示的惯例。图 2.3 (a)表示导线 1 与导线 2 完全分开，导线 1 上的信号 a 与导线 2 上的信号 b 无关。如果想连接它们，可以在交叉处加一个"点" 如图 2.3(b)所示。这时两根导线互相连接，因此在一根导线上输入一个信号 a，会使两根导线上所有各点都有同样的值。

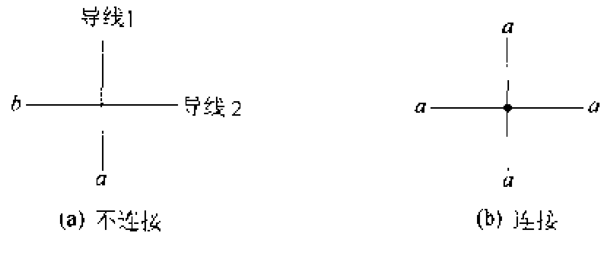


图 2.3 线路图中连接的习惯画法

让我们考察一下具有同样问题的另一个电路。在图 2.4 中，两个开关互相**并联**，由两个独立的变量分别控制。也就是说，它们的左端与左端(输入端)，右端与右端(输出端)分别连在一起。根据方程(2.2)推导出其输出的 f 值，即上面开关在 $a=1$ 且仅在 $a=1$ 时输出 $a\cdot 1$，而下面的开关在 $b=1$ 且仅在 $b=1$ 时输出 $b\cdot 1$。这些结果均在图中恰当的位置上标出。由此得出结论：如果 $a=1$ 或 $b=1$(或同时等于 1)，那么其输出可用一个表达式表示为

$$g=a+b \tag{2.5}$$

在分析中此时就是一个 OR2 操作，因此并联开关可以用来对变量进行 OR 操作。在图中用在

② 我们用 AND2 来表示两个输入的"与"(AND)操作。这种表示方式将适用于所有的门。例如 OR2 操作意为有两个输入的"或"(OR)操作。

两个开关之间加上一个"+"号来表明这一点。不过要注意，当同时 $a=0$ 和 $b=0$ 时，这一开关电路的输出 g 是不确定的，因而它不能提供一个完整的 OR2 功能。

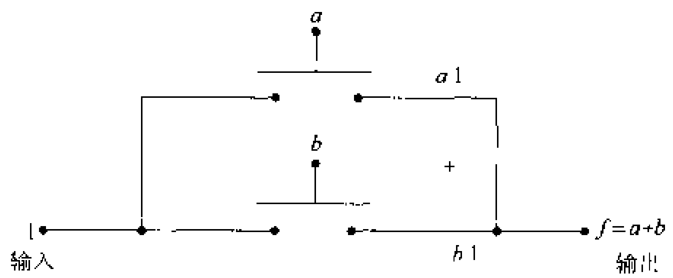


图 2.4　开关并联

前面的例子表明，开关所具有的特性可以作为实现逻辑运算的基础。既然逻辑方程(2.2)只有在开关闭合时才成立，所以是不能得到完整的"与门"(AND)和"或门"(OR)的，因为两个电路都不能产生逻辑 0 的输出。

现在，引入工作方式完全相反的另一种开关是十分有用的。这种开关叫做**低电平有效控制开关**，它定义为具有如图 2.5 所示的特点。我们在符号的上部增加一个逻辑"小圈"以区别于高电平有效控制开关。根据定义，当控制位的值为 $A=0$ 时，低电平控制开关闭合，如图 2.5 (a)所示。为了断开这一开关，必须把 $A=1$ 的值应用到这个器件上，如图 2.5(b) 所示。这一工作情形可以用逻辑方程表达为

$$y = x \cdot \overline{A} \qquad \text{iff } A = 0 \tag{2.6}$$

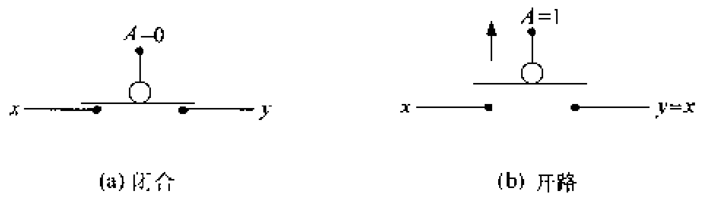


图 2.5　低电平有效开关

此时，如果 $A=1$，y 的值不确定。比较这两类开关，看到它们的工作方式是互补的。

作为如何运用这类开关的例子，考虑图 2.6 中的一对串联开关。跟踪信号由输入端通过第一个开关，得到输出值为 $\bar{a}\cdot 1$，这在且仅在 $a=0$ 时才成立。这一输出是第二个开关的输入。所以串联开关链的输出为

$$\begin{aligned} h &= (\bar{a}\cdot 1)\cdot \bar{b} \\ &= \overline{a+b} \end{aligned} \tag{2.7}$$

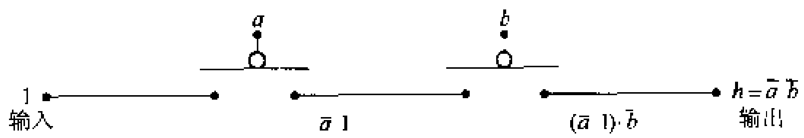


图 2.6　相补的串联开关

在这里用了 De Morgan 关系式推出式(2.7)的第二行。它看上去很像 NOR2 操作。然而由于第二个开关在 $b=0$ 时必定关闭，所以这个结果只有同时 $a=0$ 且 $b=0$ 时才成立。如果其中任一 $a=1$ 或 $b=1$，则 h 不确定。这样，我们又面临与前面例子同样的问题。

让我们进一步设想在一个电路中同时使用两种类型的开关。我们将同时输入逻辑值 1 和逻辑值 0，使对所有可能的输入组合情况下均能得到一个确定的输出值。如图 2.7 所示，高电平控制开关 SW1 用来把输入逻辑值 0 连至输出 y，而低电平控制开关 SW2 则把输入逻辑值 1 连至输出 y。变量 a 同时控制两个开关。由于两个开关并联，可把上下两个支路间的 OR 关系写成下列形式的输出：

$$y = \bar{a}\cdot 1 + a\cdot 0 \tag{2.8}$$

我们通过给 a 一个特定值来理解电路的工作情况。如果 $a=0$，则 SW1 打开而 SW2 闭合，输出为

$$y = \bar{0}\cdot 1 + 0\cdot 0 = 1 \tag{2.9}$$

如果 $a=1$，则 SW1 闭合，SW2 打开，代入式(2.8)有：

$$y = \bar{1}\cdot 1 + 1\cdot 0 = 0 \tag{2.10}$$

因此这个电路消除了不确定电压的问题。而且因为 $a\cdot 0=0$，所以表达式可简化为

$$y = \bar{a} \tag{2.11}$$

换言之，电路实现了非(NOT)操作。

$$y = \text{NOT}(a) = \bar{a} \tag{2.12}$$

这表明使用两个相反特性的开关，能够构成一个结果非常确定的电路。

图 2.7 中的非门(NOT 门)电路基于一个 2 选 1 多路选择器(MUX)的工作，如图 2.8 所示。MUX 通过控制 a 来选择输入，当 $a=0$ 时选择输入端 0(此时“1”加到 MUX 上)，或当 $a=1$ 时选择输入端 1(此时“0”加到 MUX)。输出可表示为

$$y = \bar{a}\cdot 1 + a\cdot 0 \tag{2.13}$$

简化为 $y=\bar{a}$。对图 2.7 中开关电路的仔细考察可以验证它与多路选择器是一一对应的。

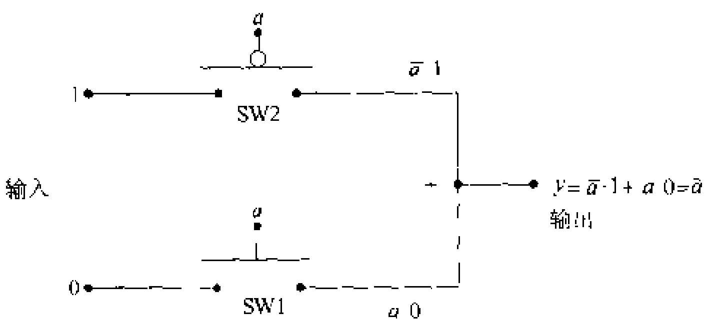


图 2.7 基于开关的非门(NOT 门)

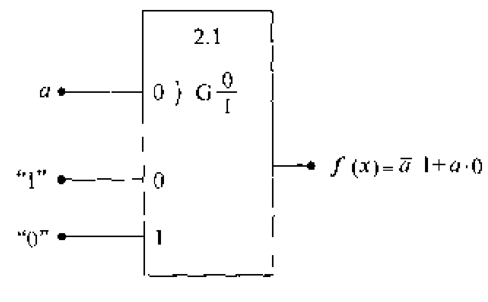


图 2.8 基于 MUX 的非门(NOT 门)

2.2 MOSFET 开关

MOSFET是在高密度数字集成电路设计中用来传输和控制逻辑信号的电子器件。[③] 缩写词 MOSFET 代表金属氧化物半导体场效应晶体管(metal-oxide-semiconductor field-effect transistor),不过现在还不必去理会这些细节。MOSFET 的工作在许多方面都非常像上一节介绍过的理想开关。但在使用之前必须了解一些重要的差别。这就是 MOSFET 必须服从电路方程,且它们的最终性能要受物理定律的限制。本节将侧重于为这些器件建立开关模型,电流方面更复杂的问题将在以后的章节中讨论。

互补 MOS(**CMOS**)采用两种类型的 MOSFET 构建逻辑电路。一种称为 n 沟道 MOSFET(或简称为 nFET),它以带负电荷的电子作为电流。nFET 的电路符号如图 2.9(a)所示。**栅极**的作用是作为器件的控制电极。在栅极上加电压就决定了在**漏**端和**源**端之间的电流。另一种晶体管称为 p 沟道 MOSFET 或 pFET,它以正电荷为电流,其电路符号如图 2.9 (b)所示。nFET 与 pFET 符号间惟一的画法差别是栅极上起反相作用的小圈。像 nFET 一样,加在 pFET 栅极上的电压决定了在源端和漏端之间的电流。不要把 MOSFET 的**栅极**与**逻辑门**搞混(英文中二者均为 gate——译者注),这两个"gate"没有关系,根据讨论的上下文总能帮助弄清它们的含义。

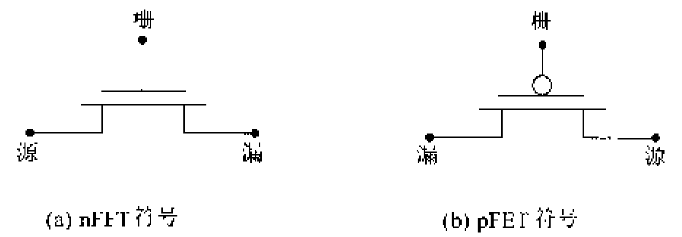


图 2.9 nFET 和 pFET 的符号

MOSFET 是真正的电子器件。在将它们用做逻辑控制开关之前,必须首先确定在布尔值与电参量之间是如何转换的。这是通过接入电源后在芯片上的电压来实现的。最一般的情况是有确定的两个电源提供电压 V_{DD} 和 V_{SS},如图 2.10 所示。参考端位于两个电源之间的**接地**连接(电压为 0 V)。这样,芯片同时得到**正**电源电压 V_{DD} 和负电源电压 V_{SS}。早期的硅 MOS 逻辑电路同时采用正、负电源电压,但现代设计仅要求一个正电压 V_{DD} 和接地连接;一般采用 $V_{DD}=5$ V 和 3.3 V 或更低。另一端的电压设为 $V_{SS}=0$,这就构成了电源供电电路见图 2.11(a)。[④] 我们将假定所有的电路都只用一个正电压电源 V_{DD}。在实际中仍

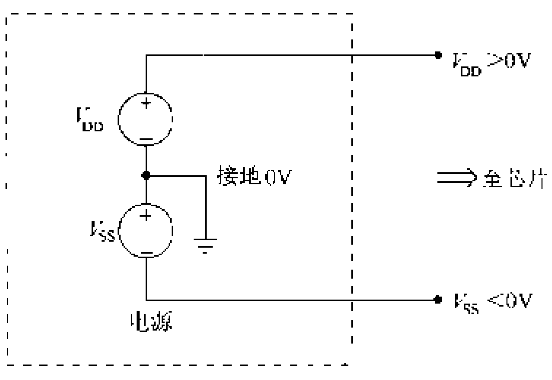
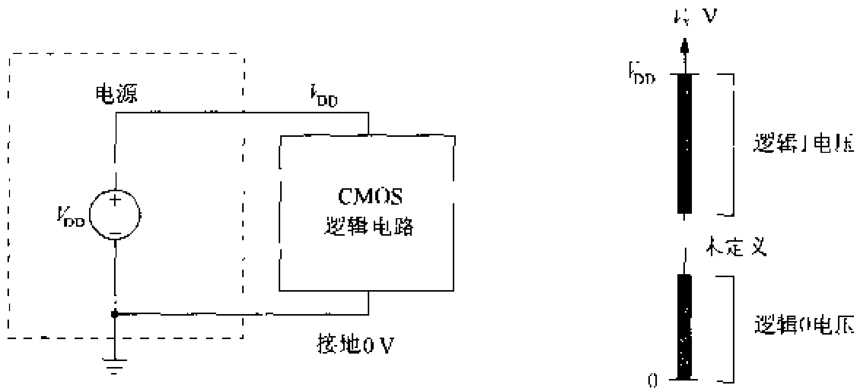


图 2.10 双电源电压

③ "MOSFET"发音为 moss-fet。

④ 本教材中电压单位伏特用 V 来表示

用 V_{SS}表示电路中的最低电压，这时 V_{SS}意指 0 V。



(a) 电源连接　　(b) 逻辑定义

图 2.11 单电压电源

现在我们可以确立逻辑值与电压间的关系了。回想一下布尔变量是离散的，一个二进制变量 x 的值只可能是 $x=0$ 或 $x=1$。在电路层次上，我们用电压 V_x 来代表变量 x，于是

$$0 \leqslant V_x \leqslant V_{DD} \tag{2.14}$$

即为电源直接提供电路 0 V 和 V_{DD}时 V_x 值的正常范围。**正逻辑惯例**于是定义**理想**逻辑值 0 和逻辑值 1 的电压为

$$\begin{aligned} x = 0 \quad & \text{意味着} \quad V_x = 0\text{ V} \\ x = 1 \quad & \text{意味着} \quad V_x = V_{DD} \end{aligned} \tag{2.15}$$

实际电路对此更为宽松，允许用一定范围的电压来代表逻辑 0 和逻辑 1，见图 2.11(b)。一般来说，

- 低电压对应逻辑值 0
- 高电压对应逻辑值 1

最高逻辑 0 电压和最低逻辑 1 电压之间的过渡区域没有定义，既不代表 0 也不代表 1。高低两个电压允许延伸的实际范围取决于逻辑电路的特性，将在以后讨论。

在定义了逻辑 - 电压间的转换后，让我们来考察一下 MOSFET 的开关特性。理想上，一个 nFET 的工作情形就像一个高电平控制开关。如图 2.12 所示，A 是应用到栅极上的逻辑变量。如果 $A=0$，相当于一个低电压，则 nFET 的作用像一个断开的开关，左右两边间不发生关系，如图 2.12(a)所示。加大栅极上的电压至一高电压值，如同改变 A，使 $A=1$。其结果使开关闭合，如图 2.12(b)所示。与高电平控制开关一样，这可以用逻辑方程表达为

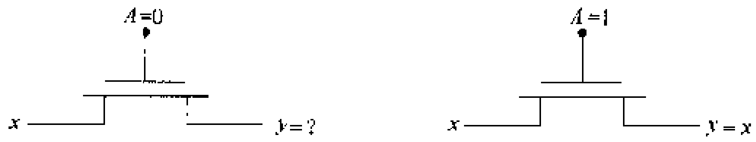
$$y = x \cdot A \tag{2.16}$$

上式在且仅在 $A=1$ 时成立。这一情况显示在图 2.12(b)中。

pFET 与之正好相反，它的工作情况像一个低电平控制开关。在图 2.13(a)中，加在栅极上的信号具有逻辑值 $A=1$，对应于一个高电压。这时，电路断开，x 和 y 之间没有直接关系。如果栅极上的电压降低到 $A=0$，则 pFET 的作用如同一个闭合的开关。这使我们可以写出如下的理想关系：

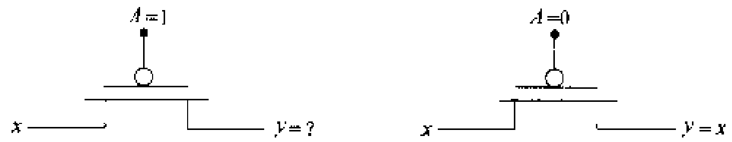
$$y = x \cdot \overline{A} \tag{2.17}$$

这一关系只要 $A=0$ 就一直成立，见图 2.13(b)。



(a) 开路　　(b) 闭合

图 2.12　nFET 开关特性



(a) 开路　　(b) 闭合

图 2.13　pFET 开关特性

MOSFET 使我们能够运用高电平控制和低电平控制开关电路技术设计逻辑电路。不过，FET 是实际的器件，不能完全像上面提到的理想开关模型那样工作。只要理解这些区别，了解它的局限性，这将不是一个严重的问题。

1. FET 阈值电压

开关方程假定应用到 FET 栅上的二进制变量 A 的值非 0 即 1，但其相对应的电压 V_A 是一个物理量，不可能如此分开。且希望给 $A=0$ 和 $A=1$ 这两种情况定义一个电压范围，以便设计工作电路。每个 MOSFET 都有一个特性参数叫做**阈值电压** V_T，它可帮助定义重要的栅电压范围。具体的 V_T 值是在生产过程中确定的，因此对于 VLSI 设计者这就是一个给定值。一个复杂的问题是 nFET 和 pFET 的阈值电压不同。

nFET 的特点是阈值电压 V_{Tn} 为正数，其典型值大约在 $V_{Tn}=0.5\text{ V}\sim0.7\text{ V}$。$V_{Tn}$ 的意义可以参照图 2.14(a)中的参数来理解。首先注意漏端表示为近电源 V_{DD} 的一端，而源端则接地(0 V)。图中的栅源电压 V_{GSn} 是决定 nFET 断开还是闭合开关的重要参数。具体来讲，如果

$$V_{GSn} \leqslant V_{Tn} \tag{2.18}$$

则晶体管的作用像一个断开的电路，在漏与源之间没有电流流过；这种情况称为晶体管**截止**(off)。反之如果

$$V_{GSn} \geqslant V_{Tn} \tag{2.19}$$

则 nFET 的漏与源接通，所对应的开关闭合。一个导电的晶体管称为**导通**(on)。这一工作情形使我们能建立如图 2.14(b)所示的电压图来确定与二进制变量 A 相应的电压 V_A。特别是，我们注意到

$$V_A = V_{GSn} \tag{2.20}$$

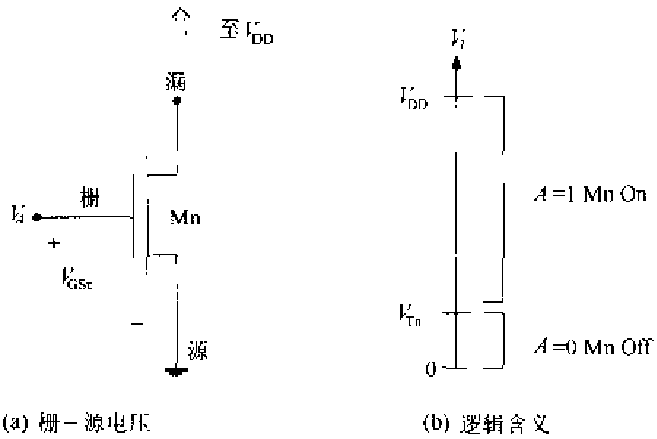


图 2.14 nFET 的阈值电压

这表明，$A=0$ 相当于 $V_A \leqslant V_{Tn}$，而 $A=1$ 则意味着 $V_A \geqslant V_{Tn}$。这些关系确定了控制 nFET 所需要的电压范围。

pFET 的行为与 nFET **互补**。考虑图 2.15(a)中的晶体管，对于 pFET，源端连至电源 V_{DD} 而漏端则近地端，这与 nFET 正好相反。在 pFET 器件中，源－栅电压 V_{SGp}是重要的外加电压。按照惯例，pFET 阈值电压 V_{Tp}是指栅－源电压 V_{GSp}，它是一个负值，其典型值的范围在 $V_{Tp}=-0.5\,V$ 到 $V_{Tp}=-0.8\,V$。本书在描述 pFET 时采用 $V_{SGp}=-V_{GSp}$，以便可用绝对值 $|V_{Tp}|$来表示阈值电压。阈值电压的意义如下：如果

$$V_{SGp} \leqslant |V_{Tp}| \tag{2.21}$$

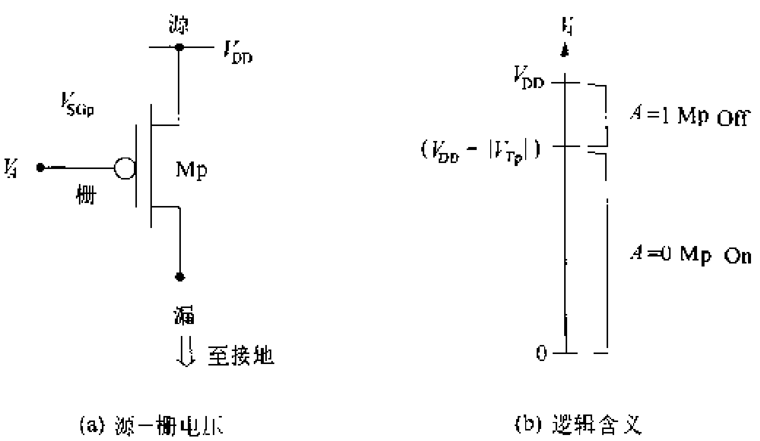


图 2.15 pFET 的阈值电压

则 pFET 截止(off)，它的作用如同一个断开的开关。相反，一个大的源－栅电压

$$V_{SGp} \geqslant |V_{Tp}| \tag{2.22}$$

使 pFET 导通(on)，其行为如一个闭合的开关。为了将这一工作情形与外加电压相联系，先要计算电压的和如下

$$V_A + V_{SGp} = V_{DD} \tag{2.23}$$

则

$$V_A = V_{DD} - V_{SGp} \tag{2.24}$$

可以看到，V_A 值低意味着 V_{SGp}大，因而 pFET 导通(on)。同样，如果 V_A 大，则 V_{SGp}小，于是 pFET 截止(off)。这就给出了逻辑 0 和逻辑 1 的范围，总结在图 2.15(b)中。注意，逻辑 0 和逻辑 1 之间的转折点为

$$V_{DD} - |V_{Tp}| \tag{2.25}$$

因为这相应于使器件导通时的源 - 栅电压。

特别要注意的是对两种类型 FET，V_A 的逻辑 0 和逻辑 1 的电压范围是不同的。解决这个问题的一种办法是，我们注意到这两种类型 FET 中无论 $A=0$ 还是 $A=1$ 值都有一部分区域重叠，在需要统一定义时可以用它们。但理想值

$$\begin{aligned} V_A &= 0\text{ V} \\ V_A &= V_{DD} \end{aligned} \tag{2.26}$$

则对两种器件都成立。

2. 传输特性

一个理想的电开关允许通过任何输入这个开关的电压。这在描述开关逻辑电路时是一个没有明确的假定，即我们所使用的开关在通过逻辑 0 和逻辑 1 时情况一样好。MOSFET 的导通能力比较有限，它不能使任意范围的电压由源通过漏，反之亦然。

让我们首先来考察一下 nFET 的传输特性。图 2.16 总结了该器件在试图将电压由左至右通过它时的工作情形。栅上的外加电压 V_{DD}保证了 nFET 导通，其作用如同一个闭合的开关。图 2.16(a)中，器件左端加上了一个逻辑 0，电压 $V_x = 0$ V，正如期望的那样，输出电压为 $V_y = 0$ V。增加输入电压时它也会被传送到输出端。但是，如图 2.16(b)所示，当加上一个理想的逻辑 1 输入电压 $V_x = V_{DD}$时，问题就发生了。这时，输出电压 V_y 的值减少为

$$V_1 = V_{DD} - V_{Tn} \tag{2.27}$$

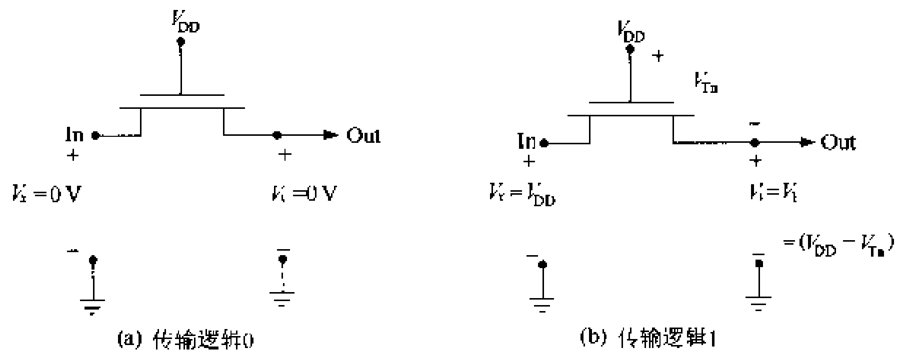


图 2.16 nFET 传输特性

它小于输入电压 V_{DD}。这称为**阈值电压损失**，它起因于保持器件的导通状态，栅 - 源电压必须具有的最小电压值

$$V_{GSn} = V_{Tn} \tag{2.28}$$

如图 2.16 所示，根据 Kirchhoff 电压定律，这要从电压 V_{DD}中减去。[5]鉴于传送电压 V_y 小于理想的逻辑 1 值 V_{DD}，称 nFET 只能导通一个“弱”逻辑 1。同理，由于它能毫无问题地产生一个输出电压 $V_y=0$ V，称它传送一个“强”逻辑 0。总之，nFET 可传送[0, V_1]范围的电压，但不能超过 V_1。

pFET 具有相反的传输特性。为了考察 pFET 特性，将它的栅极接地以加上逻辑 0。图 2.17为对应两种输入值时的电路。在图 2.17(a)中，$V_x=V_{DD}$，相当于输入逻辑 1 的情况，此时输出电压为

$$V_y = V_{DD} \tag{2.29}$$

这是理想的逻辑 1 电平。因此，pFET 能够传送强逻辑 1 电压。问题发生在试图传送理想逻辑 0 的电压 $V_x=0$ V 时，如图 2.17(b)所示。此时可传送的电压只能下降到如下的最小值

$$V_y = |V_{Tp}| \tag{2.30}$$

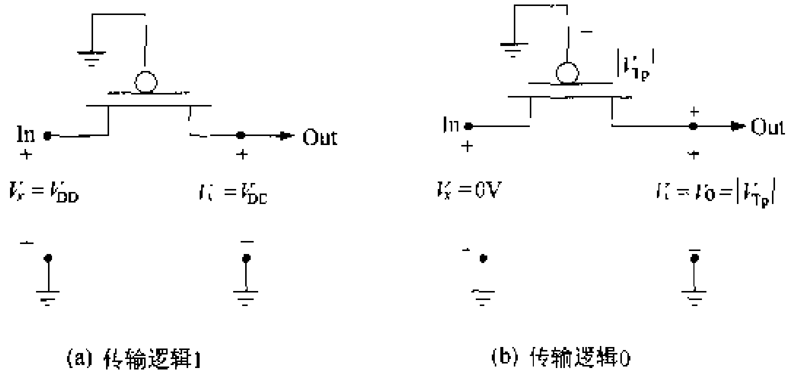


图 2.17　pFET 的传输特性

这也是由于阈值损失影响的结果。为了保持 pFET 导通所要求的源－栅电压最小值为

$$V_{SGp} = |V_{Tp}| \tag{2.31}$$

由于栅电压为 0 V，这表示源－栅电压要升高到$|V_{Tp}|$，从而影响了输出。显然，pFET 传送一个弱逻辑 0 电压。总之，pFET 传送的电压范围为[V_{DD}, V_0]，但不能低于 V_0。

让我们重新陈述一下以上讨论的结果：

- nFET 传送强逻辑 0 电压，但传送弱逻辑 1 电平
- pFET 传送强逻辑 1 电压，但传送弱逻辑 0 电平

设计互补 MOS(CMOS)电路就是为了解决传送电平的问题。具体写出以下规则作为设计的基础：

- 使用 pFET 传送逻辑 1 电压 V_{DD}
- 使用 nFET 传送逻辑 0 电压 $V_{SS}=0$ V

以上这些使我们能够构建一个可传送理想逻辑电压 0 V 和 V_{DD}到输出端的电路。不过，我们实际中将发现在并不总是需要理想的电平。

⑤ Kirchhoff 电压定律(KVL)为沿一个闭合回路电压的代数和为 0。

2.3　基本的 CMOS 逻辑门

一般 CMOS 数字逻辑门的概念可以通过图 2.18来理解。在该例子中，a，b，c 为输入位，它们联合在一起得到输出功能位 $f(a,b,c)$。按照定义这是一个数字电路，所以它所有的量都限制为 0 值或 1 值。数字逻辑电路是非线性电路，它用晶体管作为电子开关，把两个电源电压 V_{DD} 或 0 V 之一传向输出端。这对应于逻辑结果为 $f=1$或 $f=0$。在内部可将该门的输出电路看成由两个开关 SWp(低电平控制器件)和 SWn(高电平控制器件)组成。这两个开关的连接保证当一个开关闭合时另一个断开。

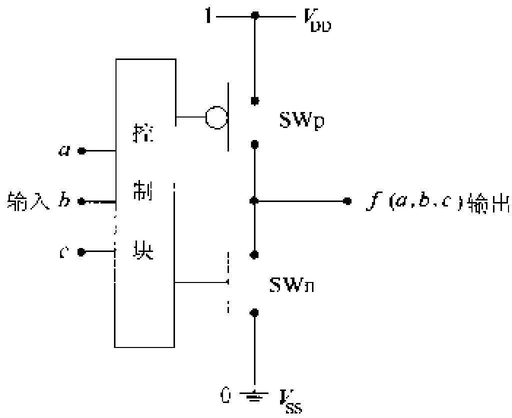


图 2.18　一般化的 CMOS 逻辑门

图 2.19 是一般逻辑门在两种可能输出情况下的工作情况。在图 2.19(a)中，上面的开关闭合而下面的开关断开，这使输出与电源连接，产生 $f=1$ 值。图 2.19(b)表示的是相反的情况，上面的开关断开，下面的开关闭合。由于这时输出端连接到 $V_{SS}=0$ V，其逻辑结果 $f=0$。虽然该图十分简化，但确实勾画出 CMOS 逻辑电路是如何工作的。该模型中惟一缺少的部分是用输入变量来控制输出开关的方法。这是由 MOSFET 来完成的。

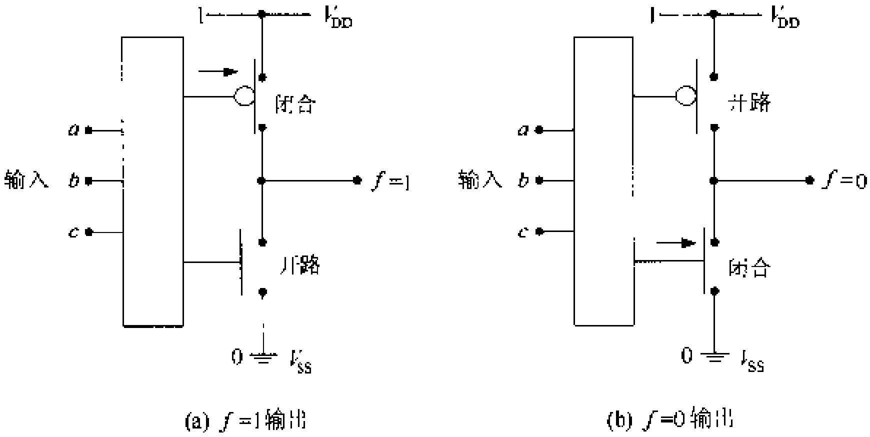


图 2.19　CMOS 逻辑门的工作情况

互补对

CMOS 逻辑电路基于用晶体管**互补对**做开关的概念。一个互补对由一个 pFET 和一个 nFET 组成，它们的栅端连在一起，如图 2.20 所示。输入信号 x 同时控制这两个 FET 的导通。注意，pFET Mp 的上面设为近电源电压 V_{DD}，而 nFET Mn 则近地(V_{SS})。通过观察图 2.21中两种可能输入值情况下每个 FET 的状态，很容易理解互补对的工作情况。在图 2.21(a)中，输入 $x=0$，使 Mp 导通而 nFET Mn 截止，其作用像一个断开的开关。

图 2.21(b)的情况相反,此时输入 $x=1$。pFET Mp 截止而 Mn 导通。互补对这个名称即来源于此:当一个 FET 导通时另一个则截止。这一工作情形的重要特点是 nFET 与 pFET 的电气特性恰好相反,这可以直接看成一种耦合的开关方式。

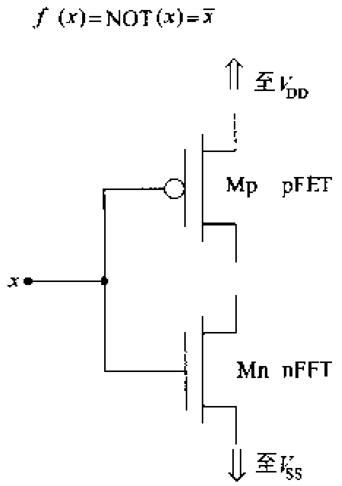
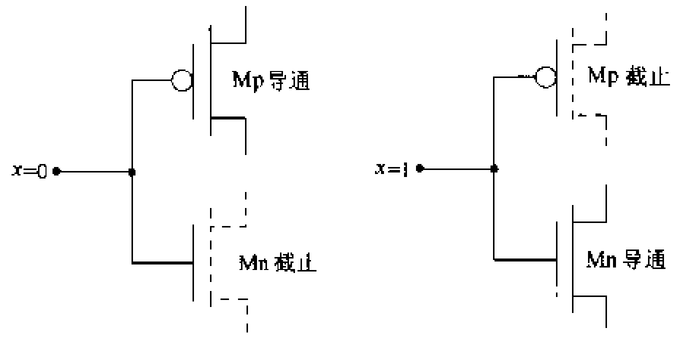


图 2.20 CMOS 互补对



(a) $x=0$ 输入 (b) $x=1$ 输入

图 2.21 互补对的工作情形

既然了解了 CMOS 逻辑门的整个结构和互补对的构想,就有了构建和分析基本逻辑门电路所需要的全部概念。

2.3.1 非门(NOT 门)

非(NOT)或反相(INVERT)函数在最简单的布尔运算中经常用到。对于一个输入 x 产生输出 $f(x)$

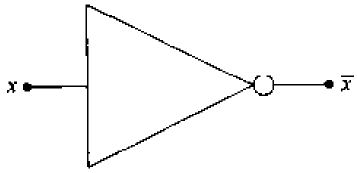
$$f(x) = \text{NOT}(x) = \bar{x} \tag{2.32}$$

即

$$\begin{aligned} &\text{若 } x = 0 \quad \text{则} \quad \bar{x} = 1 \\ &\text{若 } x = 1 \quad \text{则} \quad \bar{x} = 0 \end{aligned} \tag{2.33}$$

它定义了非门的含义。非门的逻辑符号及真值表见图 2.22,以供后用。

图 2.23 是一个 CMOS 非门。它的构建采用与图 2.7 讨论开关电路时所用的相同概念。该电路用了一个 MOSFET 互补对,输入变量 x 同时控制两个晶体管。



(a) 逻辑符号

x	$\bar{x}$
0	1
1	0

(b) 真值表

图 2.22 非门(NOT 门)

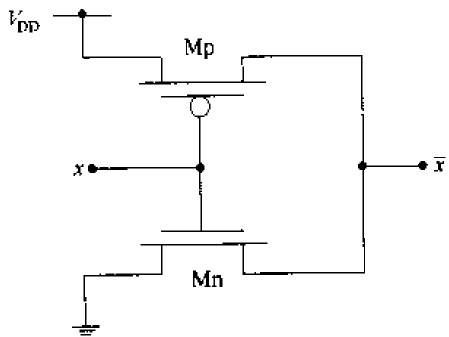


图 2.23 CMOS 非门(NOT 门)

CMOS 非门的操作直接取决于互补对的特性。如果输入值 x 为 0，则 pFET Mp 导通而 nFET Mn 截止。如图 2.24(a)所示，这使输出节点与电源电压 V_{DD}相连，使输出为 $\bar{x}=1$。反之，如果 $x=1$ 则 Mp 截止而 Mn 导通，输出于是与接地节点相连，使 $\bar{x}=0$，这可以通过图 2.24(b)的电路验证。显然，这个简单的电路确实提供了反相操作。它可以用解析的方法应用 FET 逻辑规则写出输出 f 来加以验证

$$f=\bar{x}\cdot 1+x\cdot 0 \tag{2.34}$$

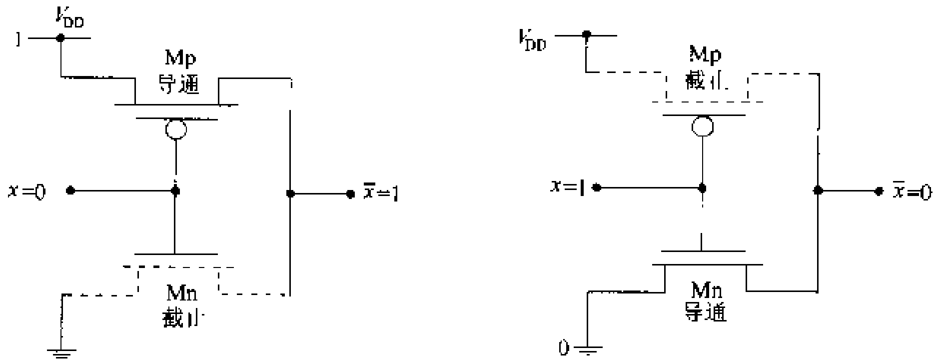


图 2.24　CMOS NOT 门的工作情况

式中，第一项代表 Mp 而第二项来自 Mn。经化简得到

$$f=\bar{x} \tag{2.35}$$

与预期的结果相符。

CMOS 非门最重要的特点之一是对于一个给定的输入状态 $x=0$ 或 $x=1$，互补 FET 对保证输出将连至 V_{DD}或者接地从而给出一个正确定义的值。尤其是这一电路避免了两个 FET 同时关闭，或两个 FET 同时打开这两种可能性。上述两种情况都会造成不正确定义的输出。

2.3.2　CMOS 或非门(NOR 门)

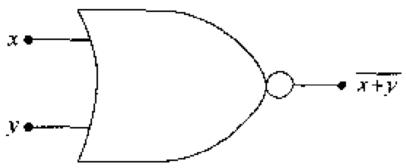
既然了解了基本的非门，我们可以扩展这一概念，利用同样的原理设计一个二输入的或非门。方法是：

- 对每个输入使用一个 nFET/pFET 互补对
- 将输出节点通过 pFET 与电源 V_{DD}相连
- 将输出节点通过 nFET 与地相连
- 确保输出总是一个正确定义的高电压或低电压

这一组设计指南可以帮助我们设计出一个输入输出特性和非门相似的逻辑电路。

NOR2 门的逻辑符号和真值表见图 2.25[⑥]。当输入变量为 x 和 y 时，NOR2 的输出为

⑥　“NOR2”意为 2 输入或非门。



(a) 逻辑符号

x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

(b) 真值表

图 2.25　NOR 逻辑门

$$g(x,y) = \overline{x+y} \tag{2.36}$$

这就是说，无论哪一个输入为 1 时 $g=0$，只有当输入组合$(x,y)=(0,0)$时才产生 $g=1$ 的结果。

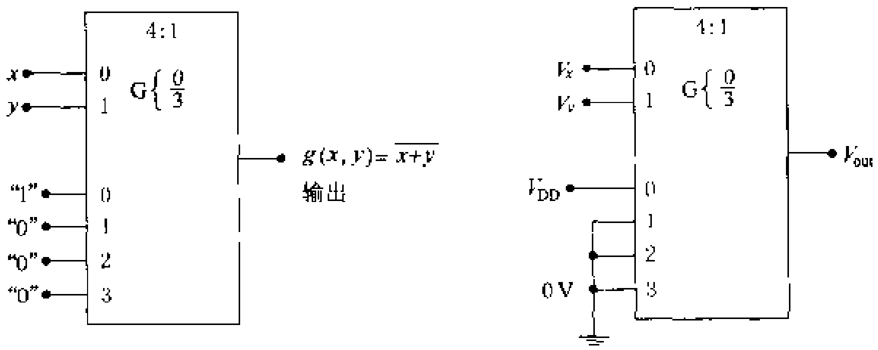
在逻辑设计层次上综合 NOR2 操作的一个方法是使用 4:1 MUX，如图 2.26(a)所示。路径通过输入对(x,y)选择，每对组合提供或 1 或 0 的输出。MUX 输出的布尔表达式为

$$g(x,y) = \bar{x}\cdot\bar{y}\cdot 1+\bar{x}\cdot y\cdot 0+x\cdot\bar{y}\cdot 0+x\cdot y\cdot 0 \tag{2.37}$$

上式应用 DeMorgan 定律可简化成预期的形式：

$$g(x,y) = \overline{x+y} \tag{2.38}$$

用电压代替二进制值可得到电压等效电路，其结果见图 2.26 (b)。图中，符号 V_x，V_y 分别代表布尔变量 x 和 y。这些内容为构建 CMOS NOR2 电路提供了基础。



(a) 逻辑方块图　　(b) 电路电压

图 2.26　用 4:1 多路选择器实现 NOR 操作

构建逻辑门的一种途径是利用图 2.27 所示的卡诺图(Karnaugh map)。由于 CMOS 门是以非门电路为基础构建的，所以它们通常产生**反相**逻辑。于是在处理卡诺图时一般会对发生的 0 和 1 都感兴趣。特别是注意到图中已得到了两个 0 一组的情形。由卡诺图可以写出以下的逻辑表达式：

$$g(x,y) = \bar{x}\cdot\bar{y}\cdot 1+x\cdot 0+y\cdot 0 \tag{2.39}$$

再回过头来构建电路。表达式中的每项代表一条连至输出的 FET 路径。第一项将输出连接到 1(电源 V_{DD})，并且通过将输入变量的反信号串联实现与(AND)来控制。第二、第三项代表

输出与 0(接地)之间两条独立的 nFET 路径。这些路径合在一起的结果就是图 2.28 中的 CMOS 二输入或非门(NOR2)电路;方程中的每项与电路的一一对应关系是十分明显的。

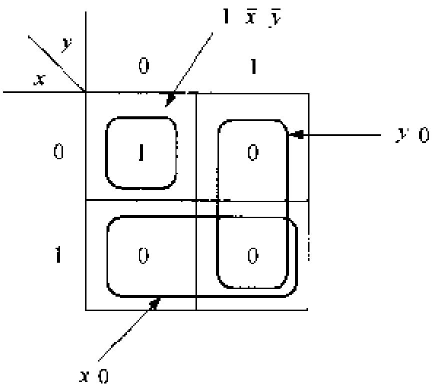


图 2.27　NOR2 门的卡诺图

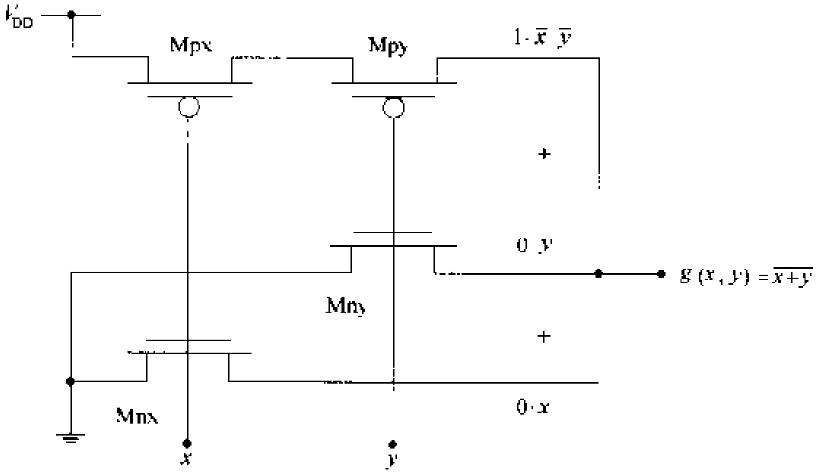


图 2.28　CMOS NOR2 门

为了验证该电路是否确实具有正确的电学功能,可以列出图 2.29 所示的表格。它包含每个 FET 在 4 种可能的输入情况下的状态(导通或截止)。跟踪每种可能的输出连接很容易看到这一开关电路与真值表是一致的。

x	y	Mpx	Mpy	Mnx	Mny	g
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

图 2.29　NOR2 门的工作情况概括

NOR2 门的电结构也表现出连接 FET 方式上的重要性。注意到两个 pFET Mpx 和 Mpy 是串联,因此它们必须同时导通才能建立从 V_{DD}到输出的导通路径。另一方面,nFET Mnx 和 Mny 是并联,所以任一 nFET 导通即可使输出接地。这被称为**串联 - 并联**晶体管连接;根据这一原理,我们可以设计出更为复杂的逻辑门。

作为一个例子,让我们以 NOR2 的连接方式为指导,构建一个 3 输入或非(NOR3)门。我们标记各输入为 x,y 和 z。每个输入连至 nFET/pFET 互补对的栅上。该门的逻辑输出表达式为:

$$f = \overline{x + y + z} \tag{2.40}$$

这就是说,如果一个或更多的输入为 1,其输出具有值 $f = 0$。由于输出为 0 是由 nFET 控制的,将三个 nFET 并联可得到正确的功能。如果应用串联 - 并联结构原理,那么三个 pFET 应当互相串联。图 2.30 即为以这种方式构成的逻辑电路;注意其与图 2.28 中 NOR2 电路的相似性。可以通过检查,验证 NOR3 逻辑电路的工作情况:如果任一输入为 1,则输出接地得到 $f = 0$;惟一能使输出产生 $f = 1$ 的情况是三个输入均为 0;这就使全部三个 pFET 都导通,同时也断开了所有的 nFET。

验证这一逻辑的另一种方法是用 FET 开关方程推导出 MUX 方程。图 2.30 中最上一条支路通过了三个 pFET 的串联组合,可表示为

$$1 \cdot \bar{x} \cdot \bar{y} \cdot \bar{z} \tag{2.41}$$

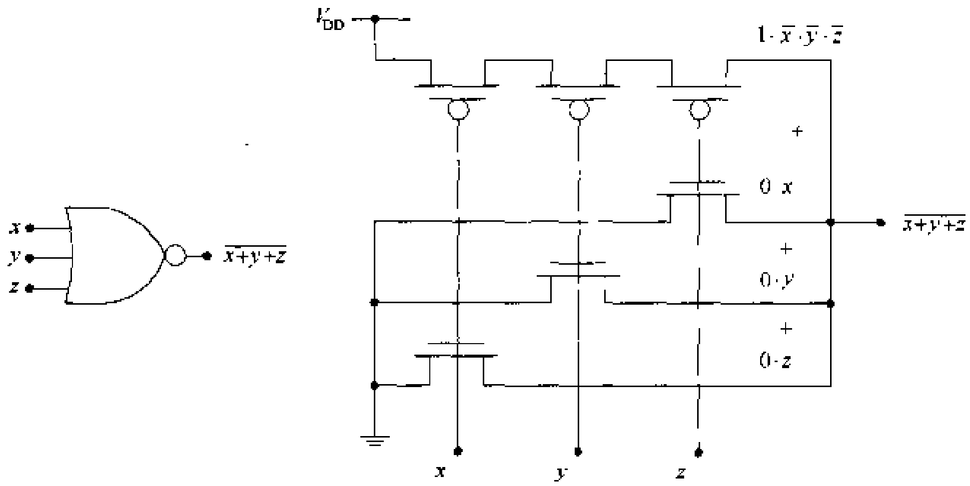


图 2.30 CMOS NOR3 门

这里,我们注意到电源电压 V_{DD}相当于逻辑 1。三条 nFET 支路中的每条都由单个 FET 组成,使输出接地。由于接地为逻辑 0,可将这四部分 OR 在一起,得到一个完整的输出表达式:

$$f = 1\cdot\bar{x}\cdot\bar{y}\cdot\bar{z}+0\cdot x+0\cdot y+0\cdot z \tag{2.42}$$

后三个 nFET 项保证了无论是一个还是多个输入为 1,该电路的输出电压为 0 V。从逻辑式看这三项均为 0 值,故上式最后变为

$$f = 1\cdot\bar{x}\cdot\bar{y}\cdot\bar{z} = \overline{x+y+z} \tag{2.43}$$

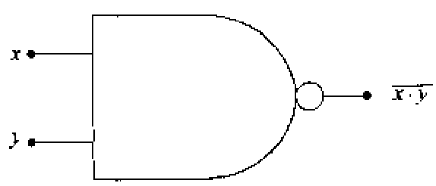
我们在简化上式时用了 DeMorgan 关系式。这个式子说明,该电路确实实现了 NOR3 操作。

从原理上讲,可用同样的方式构建 CMOS 多输入或非门,如 NOR4 或 NOR6。这一技术很容易用来构成逻辑功能电路。然而在 VLSI 的应用中,逻辑电路的选择不只是基于提供逻辑操作,还要考虑硬件特性,如开关速度以及在硅芯片上的面积消耗等。本章将仅集中于通过电路拓扑连接形成逻辑功能的问题,更多的细节考虑将在以后的章节中讨论。

2.3.3 CMOS 与非门(NAND 门)

让我们接下来构建 CMOS 二输入与非门(NAND2)电路以及它的逻辑符号和工作方式,见图 2.31。与非门的特点是除非两个输入均为 1,否则其输出为 1。由它的真值表可得到图 2.32(a)中用 4:1 MUX 来实现的这个门,其输出可表达为:

$$h(x,y) = \bar{x}\cdot\bar{y}\cdot 1+\bar{x}\cdot y\cdot 1+x\cdot\bar{y}\cdot 1+x\cdot y\cdot 0 \tag{2.44}$$



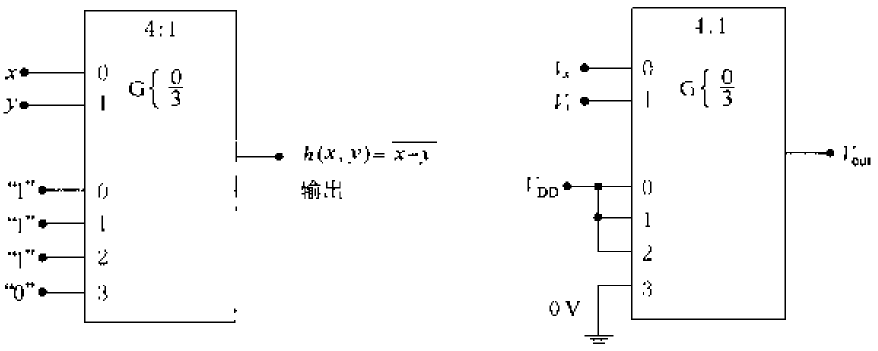
(a) 逻辑符号

x	y	$\overline{x\, y}$
0	0	1
0	1	1
1	0	1
1	1	0

(b) 真值表

图 2.31 NAND2 逻辑门

于是图 2.32(b)的电压等效电路就看得比较清楚了。



(a) 逻辑图　　(b) 电路电压

图 2.32　用 4:1 多路选择器实现 NAND2 操作

与 NOR 门一样，考察一下 NAND2 布尔函数的卡诺图很有用。图 2.33 为该卡诺图以及可用来简化 $h=1$ 情形的两组 1 的组合。利用这两个简化，表达式可改写成：

$$h(x,y) = \bar{x}\cdot 1+\bar{y}\cdot 1+x\cdot y\cdot 0 \tag{2.45}$$

把上式中的每项转化为 FET 组合得到图 2.34 的 CMOS 电路。它实现了 NAND2 功能。这可通过图 2.35 表格所概括的工作方式来验证。NAND2 门的一个重要特点是它采用两个 pFET 并联，而 nFET 则串联，这恰与 NOR2 门的结构相反。

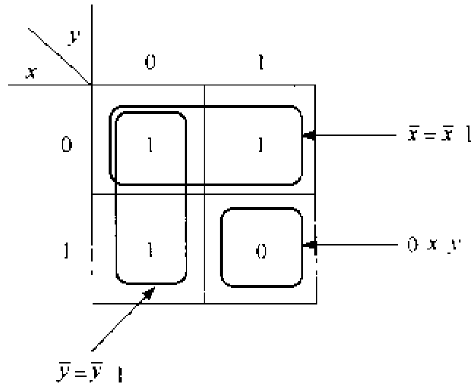


图 2.33　NAND2 卡诺图

采用同样的拓扑连接方法可以建立一个三输入与非门(NAND3)。它要求有三组互补对，每组分别由不同的输入驱动。将 nFET 串联而 pFET 并联就得到如图 2.36 所示的门。为了验证电路的功能，注意到所有三个输入必须都为 1 才能使输出与地面之间导通。如果有任一(或更多)输入为 0，则相应的 nFET 截止，而这些输入所驱动的 pFET 的作用就像一个闭合的开关，于是输出端给出一个逻辑 1 的电压 V_{DD}。

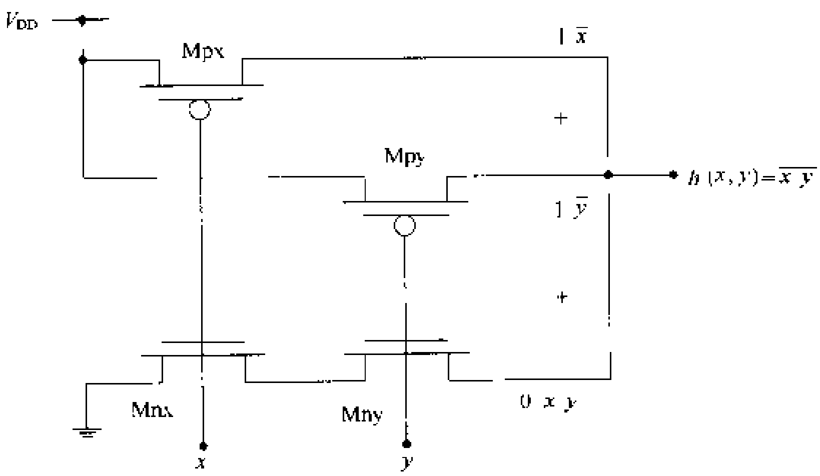


图 2.34　CMOS NAND2 逻辑电路

x	y	Mpx	Mpy	Mnx	Mny	h
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	0

图 2.35 NAND2 电路的工作情况概括

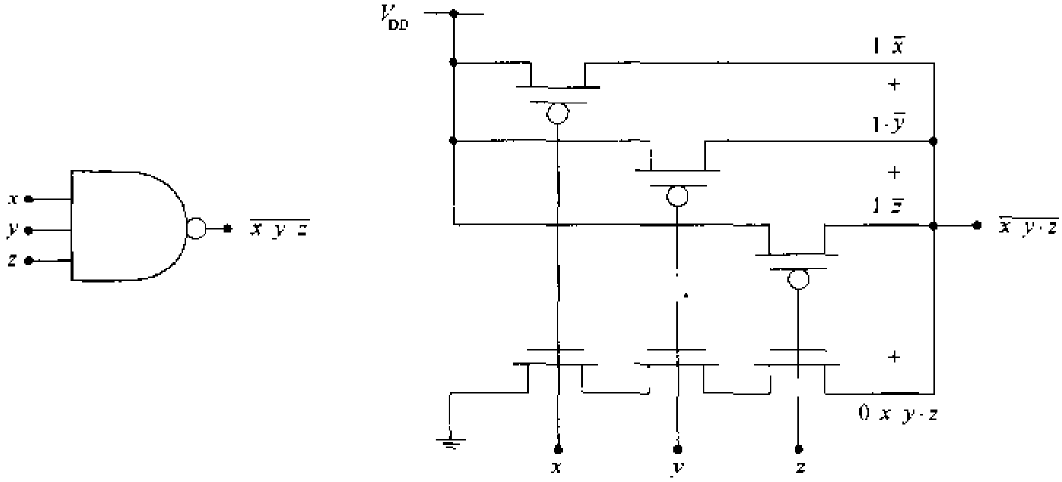


图 2.36 CMOS 中的 NAND3 逻辑门

也可以进行开关逻辑分析，即把该电路看成是一个多路选择器。在该电路下部的 nFET 串联链可用逻辑式表示为

$$0 \cdot x \cdot y \cdot z \tag{2.46}$$

每个 pFET 支路由一个晶体管组成，当输入为 0 时其作用像一个闭合的开关。四个支路间 OR 操作的结果为

$$0 \cdot x \cdot y \cdot z + 1 \cdot \bar{x} + 1 \cdot \bar{y} + 1 \cdot \bar{z} \tag{2.47}$$

消去 0 项并且运用 DeMogan 化简得到输出函数为

$$\overline{x \cdot y \cdot z} \tag{2.48}$$

即为 NAND3 功能。这一技术可以延伸来设计具有更多输入的 CMOS 与非门电路。

2.4 CMOS 复合逻辑门

用 CMOS 构成逻辑电路的最大优势之一，是能构成单个电路实现几个最基本的逻辑操作(非、与、或)的组合。在我们的讨论中，它们被称为**复合门**或**组合逻辑门**。复合逻辑门在 VLSI 系统级的设计中十分有用。

为了说明复合逻辑门的主要思想，考虑以下的布尔表达式

$$F(a, b, c) = \overline{a \cdot (b + c)} \tag{2.49}$$

构造具有这个功能的逻辑电路的最简单方法是用一个或门，一个与门和一个非门，如图 2.37(a)所示。如果可以采用一个二输入与非门的话，还可以把它简化为图 2.37(b)所示的电路。若要建立上面任何一个的等效电子电路，那么传统的办法是采用一一对应的映射：每个门需要一个电子逻辑电路。对第一种情况即图 2.37(a)就会需要三个独立的门，而图 2.37(b)使门的数目减为两个。在许多应用中，这一方法完全适用，它直观且可以直接实现。

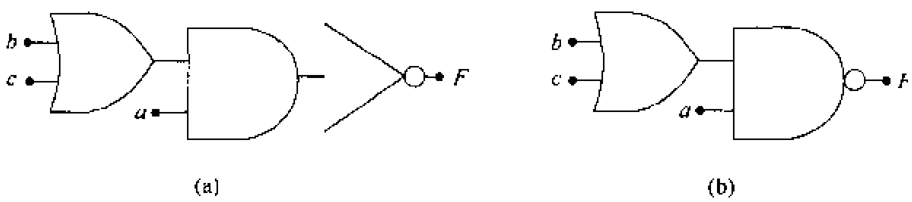


图 2.37　逻辑功能举例

VLSI 设计的约束较难满足。晶体管在硅芯片上占据面积，每个逻辑门都要用到晶体管。由于一个 VLSI 芯片上门的数量很容易超过数十万个，所以经常寻求既能完成要求的逻辑功能又能减少门和(或)FET 数目的技术。在目前的讨论中将通过建立能够完成全部功能的单个门来达到这个目的。

让我们更仔细地研究一下函数 F，即对这个函数运用 DeMogan 展开式得到

$$\begin{aligned} F &= \overline{a\cdot(b+c)} \\ &= \bar{a}+\overline{(b+c)} \\ &= [\bar{a}+(\bar{b}\cdot\bar{c})]\cdot 1 \end{aligned} \tag{2.50}$$

最后一步只是把结果与逻辑 1 相与(AND)。展开得到

$$F = \bar{a}\cdot 1+(\bar{b}\cdot\bar{c})\cdot 1 \tag{2.51}$$

上式的形式使我们可以构建一个如图 2.38 所示的 pFET 开关电路。通过检查每一项，可以验证其对应性。第一项表示一个 pFET 连接在电源(V_{DD})和由输入 a 控制的输出之间。第二项与 NOR2 门中遇到过的形式完全一样，它表示两个串联的 pFET(其控制变量为 b 和 c)把电源连至输出端。

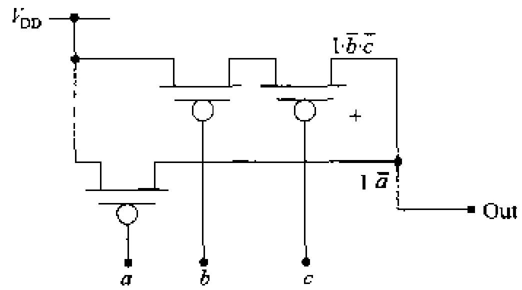


图 2.38　实现方程(2.51)输出 F 的 pFET 开关电路

仅有 pFET 电路还不足以构建实现 F 功能的电子电路。还必须加入 nFET 阵列以在需要时能使 $F=0$。方程(2.49)中函数的原始形式表明 $F=0$ 发生在

$$当\ a=1\ \text{AND}\ (b+c)=1\ 时$$

这等同于输出表达式

$$0\cdot[a\cdot(b+c)] \tag{2.52}$$

上式又可以用来描述图 2.39 中 nFET 阵列。两个由 b 和 c 控制的并联 nFET 实现 OR 操作。

这一并联组合又与输入为 a 的一个 nFET 串联产生 AND 操作。其逻辑可通过图 2.40 所示的卡诺图 0 项组合简化来验证。之所以可用输入为 a 的单个 nFET 来简化，是因为在卡诺图中一个公共项被两个组合所包含。

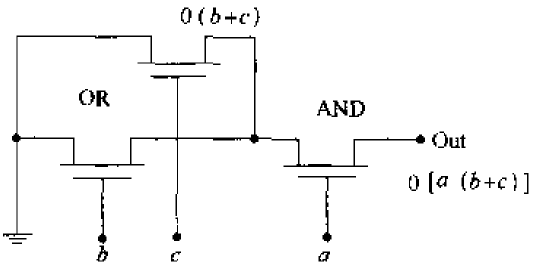


图 2.39 实现 F 的 nFET 逻辑电路

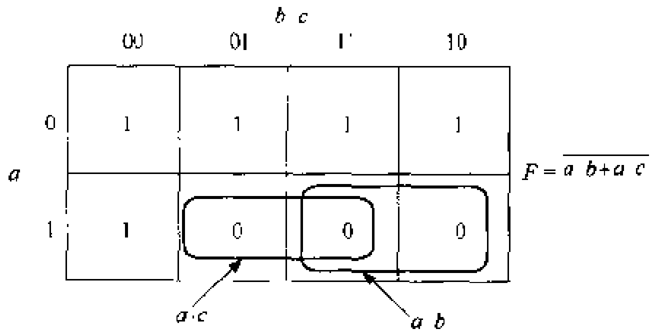


图 2.40 nFET 电路卡诺图的组合简化

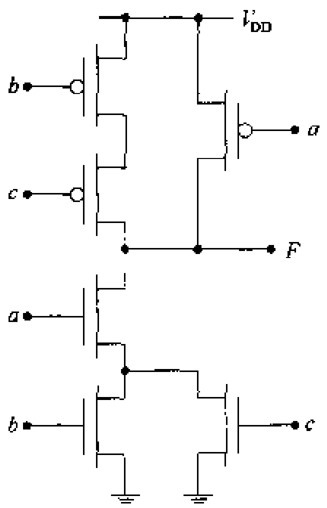


图 2.41 最终完成的 CMOS 复合逻辑门电路

完整的 CMOS 逻辑门是将 nFET 和 pFET 组合在一起，形成像图 2.41 所示的电路。将 FET 的方向旋转 90°，即为最终完成的线路图。这是最常用的画 CMOS 逻辑电路的方法，因为它使串联和并联的 FET 更为清楚。电路的等效性可以通过考察每个支路且与上面介绍的简单电路比较来验证。

这个例子表明，一个复杂的功能可由单个 CMOS 逻辑电路来实现，它代替了由两个或多个最基本门的串接。复合逻辑门电路在 VLSI 设计中可以更为有效，这是由于它们简化了电路要求和逻辑信号流。CMOS 技术功能强的一个特点，是它使我们能够利用几种不同的技术，如复合逻辑门来设计逻辑电路。这有助于加大**集成密度**，它衡量在一个硅芯片上可以包含的逻辑量。

2.4.1 结构化逻辑设计

复合逻辑门的结构化设计方法可以通过注意电路特性而得到启发。CMOS 逻辑门本质上是**反相**的，这就是说输出总是产生一个与输入变量反相的操作。图 2.42中简单的反相器说明了这一性质的来源。如果输入 a 是逻辑 1，则 nFET 导通而 pFET 截止。nFET 将逻辑 0(接地)传送到输出端，于是在那里得到 $\bar{a}$。这一特性在与非和或非电路中也能观察到。

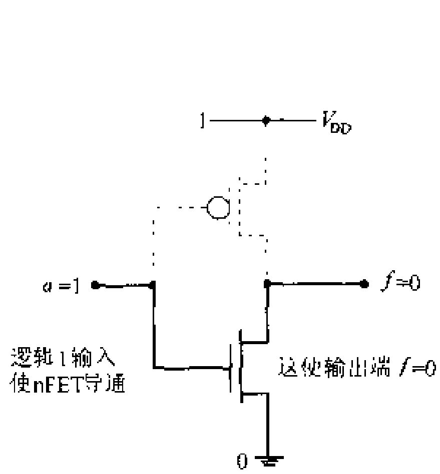


图 2.42 CMOS 门反相特性的来源

CMOS 逻辑电路的这种反相本质使我们得以运用结构化方法构造 AOI 和 OAI 逻辑表达式表示的逻辑电路。AOI 逻辑功能是按先“与”(AND)后“或”(OR)

再后“非”(NOT 即反相)的顺序执行操作。例如,

$$g(a,b,c,d)=\overline{a\cdot b+c\cdot d} \tag{2.53}$$

意味着如下的操作顺序:先进行

$(a \text{ AND } b)$和$(c \text{ AND } d)$

然后进行或(OR)操作,其最后结果为

$g=\text{NOT}[(a \text{ AND } b)\text{OR}(c \text{ AND } d)]$

另一个例子是将前面 CMOS 门的函数展开

$$f(a,b,c)=\overline{a\cdot(b+c)}=\overline{a\cdot b+a\cdot c} \tag{2.54}$$

在展开各项后可以看到这是一个 A-O-I 的操作顺序。换一种说法,AOI 是将积的和(SOP, sum-of-products)反相。OAI(OR-AND-INVERT)则将“与”和“或”的操作顺序反一下。其形式如以下例子

$$h(x,y,z,w)=\overline{(x+y)\cdot(z+w)} \tag{2.55}$$

上式意味着首先计算

$(x \text{ OR } y)$和$(w \text{ OR } z)$

然后计算

$h=\text{NOT}[(x \text{ OR } y)\text{AND}(w \text{ OR } z)]$

即可求出 h 的值。一个 OAI 形式等同于一个和的积(POS)反相的表达式。

CMOS 的开关特性为实现像 AOI 和 OAI 这样的反相逻辑形式提供了很自然的方法。这个技术的基础是以一种一致的方式运用 nFET 和 pFET。这种复合逻辑门使设计者能够将三个或更多的基本操作压缩到一个逻辑门中。首先考虑 nFET 形成逻辑的特点。从与非门(NAND)的分析中可以看到,串联 nFET 提供“与 - 非”逻辑,见图 2.43(a);类似地,或非门的分析表明,并联的 nFET 产生“或 - 非”操作,如图 2.43 (b)所示。这些结论可以归纳应用到多个晶体管。例如,输入为 a,b,c,d 的四个串联 nFET 将产生

$$\overline{a\cdot b\cdot c\cdot d} \tag{2.56}$$

而并联的 FET 将产生“或 - 非”操作

$$\overline{a+b+c+d} \tag{2.57}$$

从这一观察得到的好处是可以将串联和并联 nFET 组合起来产生复合逻辑门。图 2.44 就是这样的一个例子。这个阵列由并联的两组 FET 组成,每个组含有两个串联的 nFET。左边的一组晶体管产生“与”操作$(a\cdot b)$,而右边的一组晶体管产生“与”操作$(c\cdot d)$;左右两组并联形成“或”操作,而该门的最终输出产生了“非”操作。由此可见,该功能可表达为

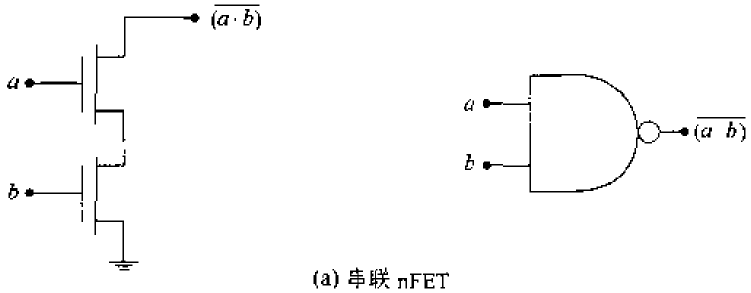
$$X=\overline{(a\cdot b)+(c\cdot d)} \tag{2.58}$$

上式是一个 AOI 表达式,可用电路图右边的逻辑电路表示。应当注意,“非”操作是发生在这一逻辑的输出处(即仅对函数 X)。“与”操作由串联的 nFET 提供,而“或”操作是通过并联组来完成的。虽然这一方法以观察逻辑的形成为基础,但可以用开关方程的形式来验证其结果。

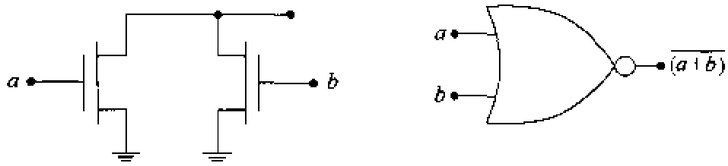
应用 nFET 方程可得输出为:

$$0\cdot[(a\cdot b)+(c\cdot d)] \tag{2.59}$$

上式与 X 的表达式等同。



(a) 串联 nFET



(b) 并联 nFET

图 2.43　由 nFET 构成逻辑

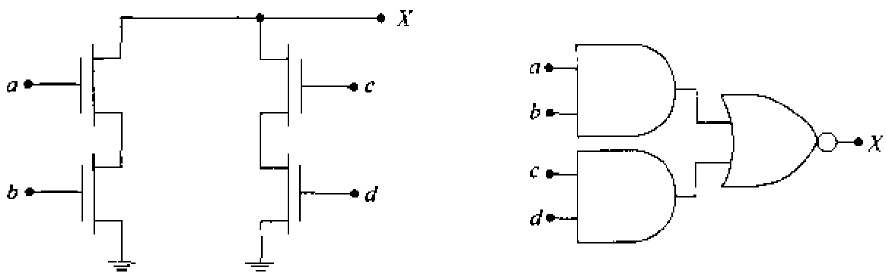


图 2.44　nFET AOI 电路

图 2.45 是一个修改过的电路。与图 2.43 比较可以看到,它增加了一条连接。现在上面的两个晶体管(输入分别为 a 和e)互相并联,类似地输入为 b 和 f 的两个 nFET 也并联。这两组并联都实现“或”操作,得到($a+e$)和($b+f$)两项。它们的串联产生“与”操作,所以反相的输出结果为

$$Y=\overline{(a+e)\cdot(b+f)} \tag{2.60}$$

上式是 OAI 形式。为了验证这一结果,采用开关级方程写出

$$0\cdot[(a+e)\cdot(b+f)] \tag{2.61}$$

它与方程(2.60)中 Y 的表达式等同。

现在回想一下,CMOS 逻辑门用 nFET 向输出传递 0,用 pFET 传递逻辑 1。由于 pFET 与 nFET 互补,可以总结出它们形成逻辑的特点,如图 2.46 所示。图 2.46(a)中并联的 pFET 可用如下逻辑方程表示

$$1\cdot(\bar{x}+\bar{y})=1\cdot\overline{(x\cdot y)} \tag{2.62}$$

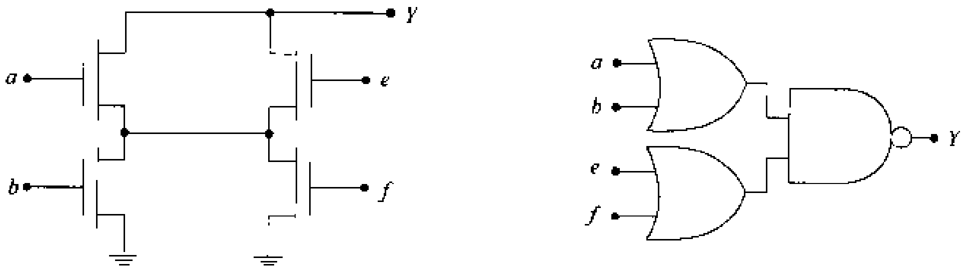
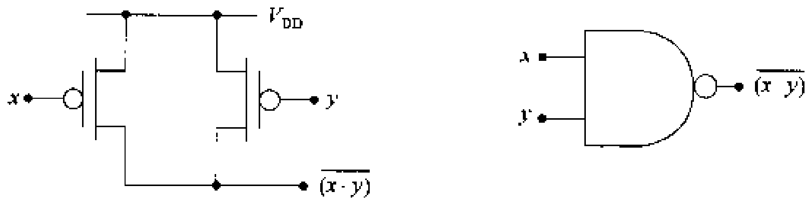


图 2.45　nFET OAI 电路

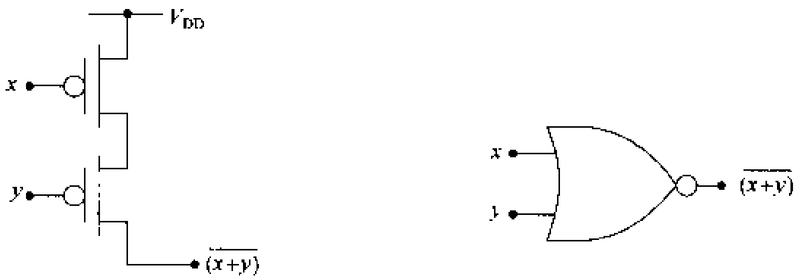
上式即为“与－非”操作顺序。为了获得“或－非”操作，必须使用串联 pFET，如图 2.46(b)所示。这时，由开关方程可得到逻辑式

$$1 \cdot \bar{x} \cdot \bar{y} = 1 \cdot \overline{(x+y)} \tag{2.63}$$

该式验证了以上陈述。



(a) 并联 pFET



(b) 串联 pFET

图 2.46　由 pFET 构成逻辑

让我们考察一下为实现前面讨论过的图 2.44 nFET 电路的 AOI 功能，即

$$X = \overline{(a \cdot b)+(c \cdot d)} \tag{2.64}$$

所需要的 pFET 阵列。利用 pFET 规则得到如图 2.47(a)所示的电路。类似地，OAI 功能

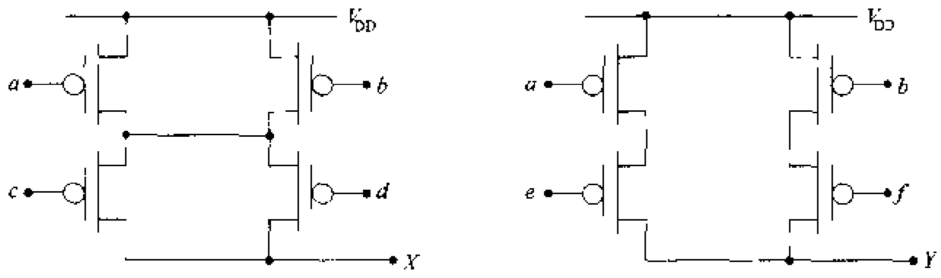
$$Y = \overline{(a+e)\cdot(b+f)} \tag{2.65}$$

产生了如图 2.47(b)所示的 pFET 阵列。

以上讨论表明，nFET 和 pFET 组合形成的工作方式不同。并联 nFET 产生“或－非”操作，而并联 pFET 则给出“与－非”操作顺序；串联 nFET 给出“与－非”，而串联 pFET 则给出

"或－非"。根据这些结果我们可以说,连接相同的两组 nFET 和 pFET 在逻辑上互为对偶。换言之,如果一组 nFET 产生下列功能:

$$g = \overline{a \cdot (b + c)} \tag{2.66}$$



(a) pFET AOI 电路 (b) pFET OAI 电路

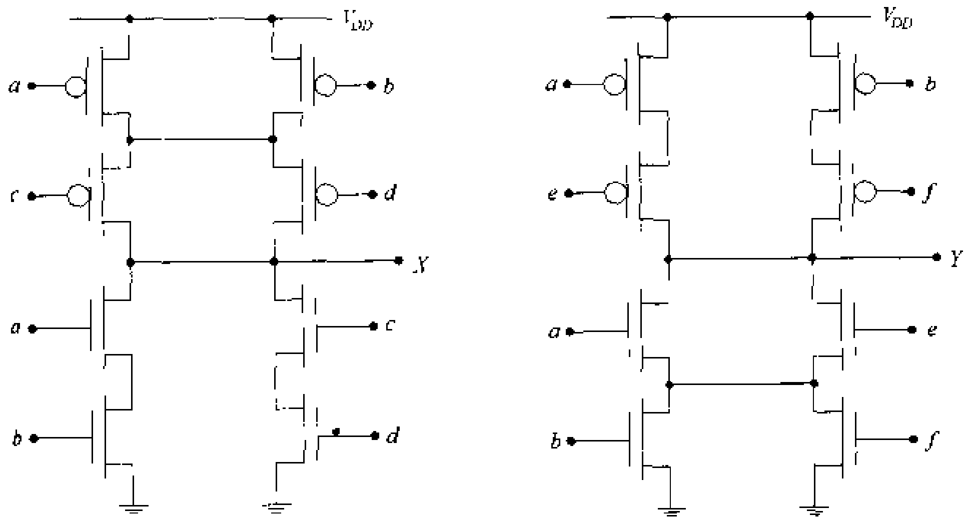
图 2.47 构成 AOI 和 OAI 门的 pFET 阵列

则连接相同的一组 pFET 阵列给出其对偶功能:

$$G = \overline{a + (b \cdot c)} \tag{2.67}$$

这里,"与"操作和"或"操作进行了互换。这一 nFET－pFET 逻辑的有趣特点可以用在一些 CMOS 的设计中。

这些例子的最重要之处可以通过将它们每个都构成一个完整的 CMOS 电路来体现,如图 2.48 所示。首先考虑图 2.48(a)的 AOI 电路。输入为 a 和 b 的 nFET 串联,而对应的 pFET 并联。输入变量为 c 和 d 的 FET 也同样如此。最后,输入为(a, b)的一组 nFET 与输入为(c, d)的一组并联,所以对应的 pFET 组是串联的。这是 nFET-pFET 阵列串－并结构的又一个例子。图 2.48(b)中的 OAI 电路也有同样的特点。这时,输入为 a 和 e 的 nFET 以及输入为 b 和 f 的 nFET 都并联,而输入为 a 和 e 以及输入为 b 和 f 的 pFET 则都串联。最后,nFET(a,e)组与(b,f)组串联,所以对应的 pFET 组就是并联。这可用来在 CMOS 中构建任意的 AOI 或 OAI 电路。



(a) AOI 电路 (b) OAI 电路

图 2.48 完整的 CMOS AOI 和 OAI 电路

【例 2.1】 考虑如下复合功能

$$X = \overline{a + b \cdot (c + d)} \tag{2.68}$$

它的 nFET 电路可按以下方式构成。

第一组:输入为 c 和 d 的 nFET 并联;

第二组:输入为 b 的一个 nFET 与第一组串联;

第三组:输入为 a 的一个 nFET 与“第一组－第二组”的电路并联。

图 2.49 中的电路清楚地显示了上述各组。pFET 采用串并联结构连接。每组 pFET 与具有同样输入的一组 nFET 相对应,即

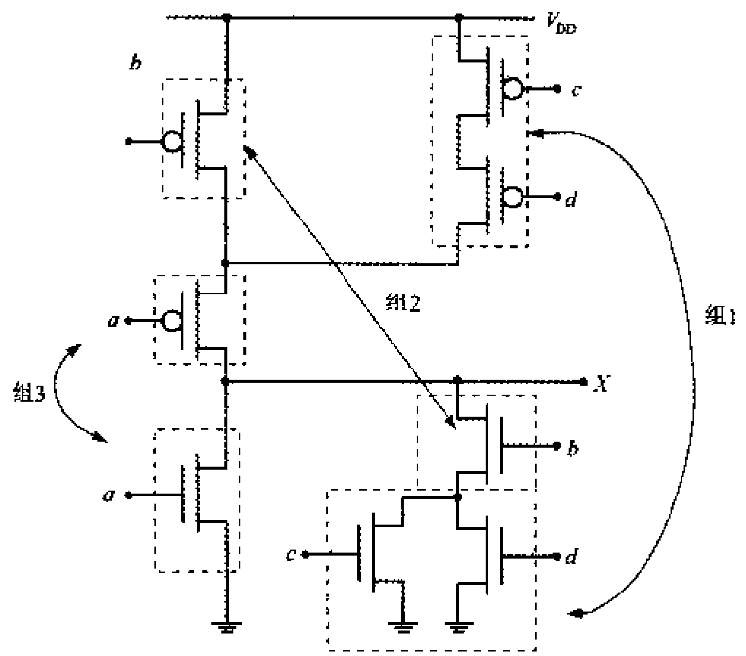


图 2.49　例 2.1 中的 AOI 电路

第一组:输入为 c 和 d 的 pFET 串联;

第二组:输入为 b 的一个 pFET 与第一组 pFET 并联;

第三组:输入为 a 的一个 pFET 与“第一组－第二组”的 pFET 串联。

与该电路等效的逻辑图见图 2.50。由输入到输出跟踪数据流可以看出,该门具有 OAOI 结构。这实际上就是一个 AOI 电路外加一个 OR 输入。

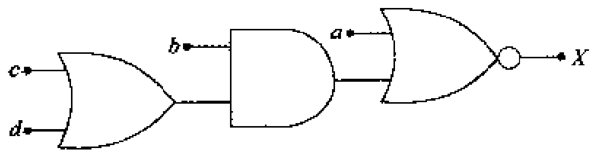


图 2.50　例 2.1 的等效逻辑图

移动反相小圈的方法

复合 CMOS 逻辑电路的串－并联布线，可以通过一种基于逻辑图的方法来设计。这个方法是将 DeMorgan 规则应用于 pFET 的连接关系，如图 2.46 所示。回想一下，pFET 可以看成一个低电平控制开关模型。因此，可把 pFET 组看成具有低电平控制输入的逻辑门。于是得到如图 2.51 所示修改后的逻辑关系。在图 2.51(a)中应用 DeMorgan 规则可写成

$$1\cdot(\overline{x\cdot y}) = 1\cdot(\bar{x}+\bar{y}) \tag{2.69}$$

所以并联的 pFET 可以看成一个具有低电平控制(带小圈)输入的或(OR)操作。以同样方式，图 2.51(b)中串联的 pFET 提供具有低电平控制输入的与(AND)操作，这可以由下列等式得到验证

$$1\cdot(\overline{x+y}) = 1\cdot(\bar{x}\cdot\bar{y}) \tag{2.70}$$

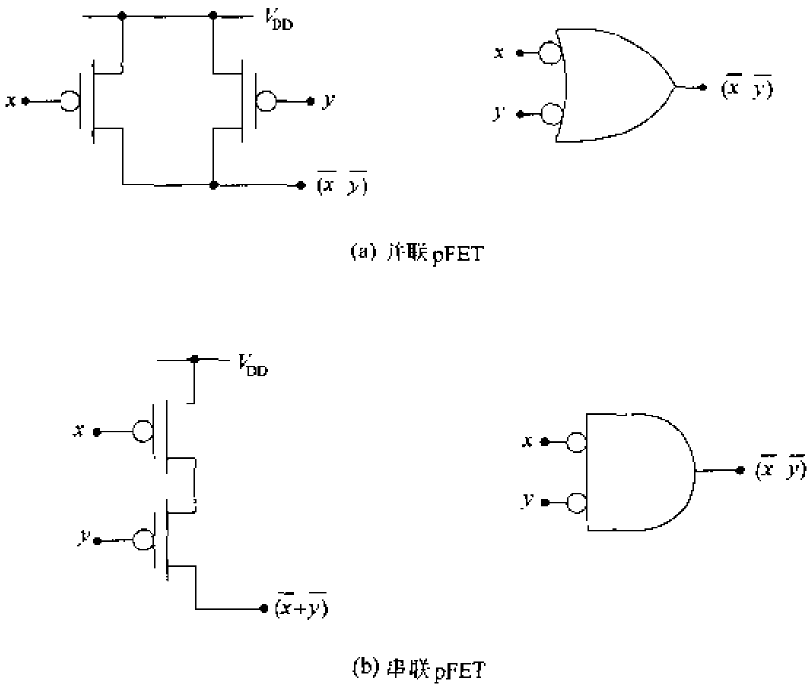


图 2.51 pFET 的低电平控制模型

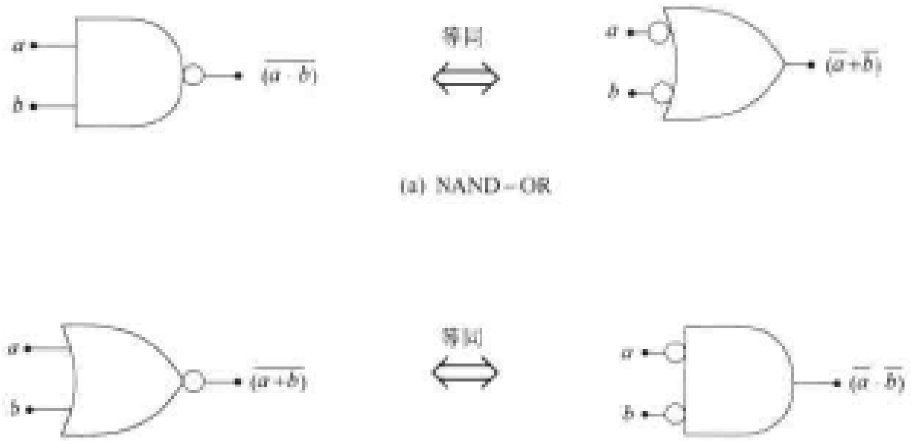
这两个操作都可以用图 2.52 所示的操作来表示。可以看到，把小圈往回移过逻辑门至输入端产生了具有低电平控制输入端的对偶操作。

CMOS 逻辑门的晶体管电路设计顺序可概括为以下步骤。

- 用基本的 AOI 或 OAI 结构构成逻辑图。允许如 OAOI 和 AOAI 这样的深层嵌套。
- 用图 2.43 中总结的逻辑门与 nFET 的关系，在输出与地之间构成 nFET 逻辑电路。
- 为了得到 pFET 阵列的连接拓扑，首先从原始的逻辑图开始，运用 DeMorgan 规则将小圈推回输入端。继续往回推使每个输入都加上小圈。利用图 2.51 中的规则即可得到在输出与 V_{DD} 之间的 pFET 电路。

注意，nFET 和 pFET 都要连接，使并联的晶体管实现或(OR)操作，而串联的 FET 实现与

(AND)操作。二者之间的惟一差别是 nFET 是高电平控制器件，而 pFET 是低电平控制(输入端带小圈)的开关。



(b) NOR-AND

图 2.52　运用 DeMorgan 规则移动小圈

【例 2.2】　考虑图 2.53 的逻辑图。这是一张构造 nFET 逻辑阵列的图。可以看到，输入为 a 和 b 的 nFET 及输入为 c 和 d 的 nFET 分别串联在一起(因为是与门)。这两个串联组与一个输入为 e 的 nFET 并联，因为在输出端它们被"或"在一起。在 nFET 阵列中或非门的非(NOT)操作是自动完成的。

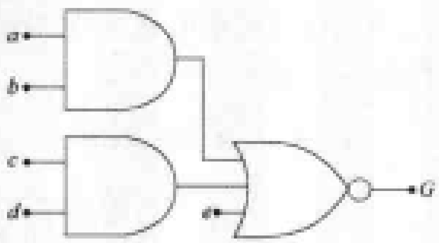


图 2.53　移动小圈例子中的 AOI 逻辑图

为了得到pFET布线，可以如图5.54所示将小圈回推。第一步是将输出或非门转变为一个具有低电平控制输入的与门，其结果为如图 2.54(a)所示的中间图。继续将小圈回推通过与门得到图 2.54(b)的低电平控制或门。这表明该 pFET 阵列的组成为

- 两个输入为 a 和 b 的 pFET 并联
- 两个输入为 c 和 d 的 pFET 并联
- 一个输入为 e 的 pFET 与以上两组串联

最终的电路如图 2.55 所示。值得去跟踪一下该电路的构造过程。同时注意到该 CMOS 逻辑门实现了逻辑图表现的 G 的全部功能。不可能把该电路再分割为更基本的逻辑。

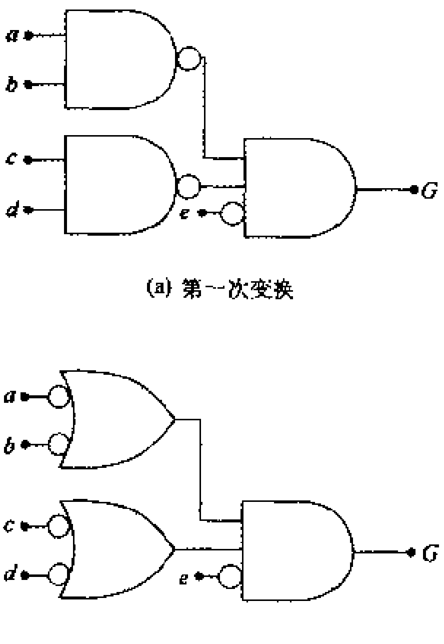


图 2.54 移动小圈得到 pFET 阵列的连接

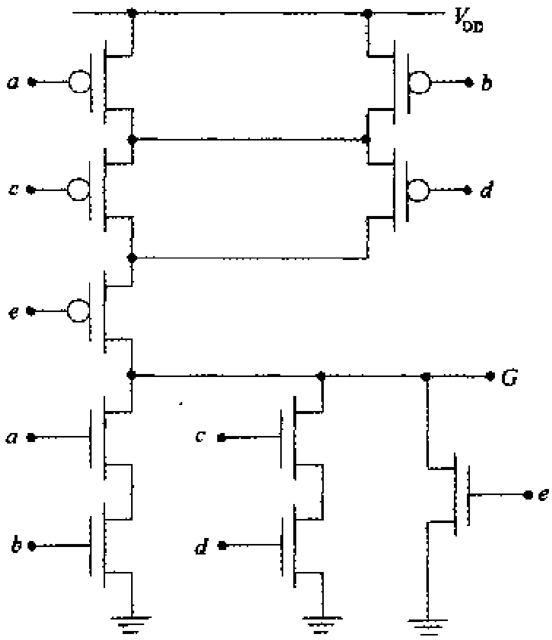
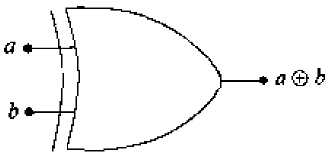


图 2.55 移动小圈例子的最终电路图

2.4.2 异或门(XOR)和异或非门(XNOR)

运用 AOI 电路的一个重要例子是建立"异或"(XOR)和"异或非"(XNOR)电路。这些常用门是由基本逻辑构成的。图 2.56 为异或门的电路符号和真值表。读出表中输出为逻辑 1 的第二、第三行得到标准 SOP(积的和)方程

$$a \oplus b = \bar{a} \cdot b + a \cdot \bar{b} \tag{2.71}$$



a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

图 2.56 异或(XOR)门的符号与真值表

这不是 AOI 形式。如果读出输入为 0 的两行,则得到异或非(XNOR)的表达式为

$$\overline{a \oplus b} = a \cdot b + \bar{a} \cdot \bar{b} \tag{2.72}$$

于是 XOR 可表示为

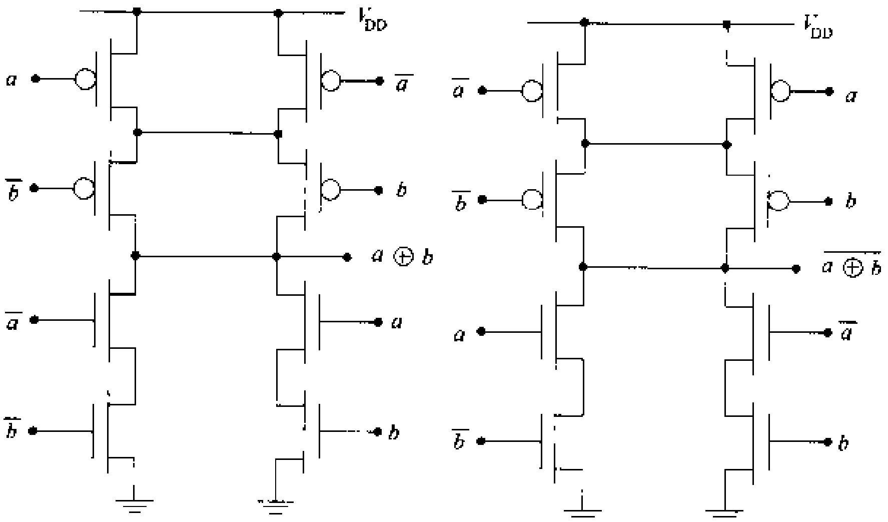
$$a \oplus b = \overline{(\overline{a \oplus b})} = \overline{a \cdot b + \bar{a} \cdot \bar{b}} \tag{2.73}$$

这是 AOI 结构。运用图 2.48(a)的电路可得到图 2.57(a)所示基本的 AOI 异或电路。由于异或门的输入只有(a,b),该电路还需要两个反相器来提供一个 4 输入$(a, b, \bar{a}, \bar{b})$组。

为了得到异或非电路，只需要求异或 SOP 方程的反，得到

$$\overline{a \oplus b} = \overline{\bar{a} \cdot b + a \cdot \bar{b}} \tag{2.74}$$

互换异或电路中的 a 和 $\bar{a}$ 即得到图 2.57(b)的异或非门。互换 b 和 $\bar{b}$ 变量也可以得到同样的结果。

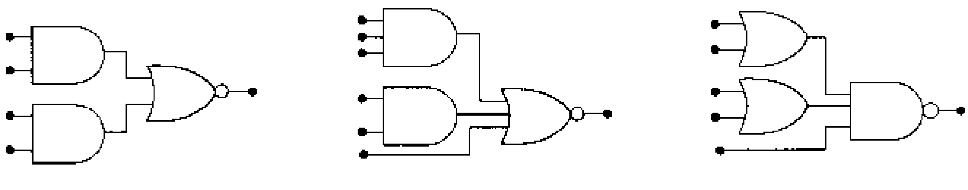


(a) 异或(XOR)　　(b) 异或非(XNOR)

图 2.57　AOI XOR 和 XNOR 门

2.4.3　一般化的 AOI 和 OAI 逻辑门

标准逻辑设计经常使用一般化的多输入 AOI 和 OAI 逻辑门来简化。这对于以预先设计的逻辑电路为基础的 ASIC 型设计更是如此。图 2.58 是对各种不同输入形态的一个简单的分类命名。图 2.58(a)中的电路是一个具有两个二输入与门的 AOI 样式，它因此称为 AOI22 门。同样，图 2.58(b)中的逻辑样式称为 AOI321 门，这里"1"表示有一个输入绕过与门直接连到或门上。图 2.58(c)按同样的约定称为 OAI221 门。采用串－并联布线或小圈推移法很容易设计相应的 CMOS 电路。



(a) AOI22 门　　(b) AOI321 门　　(c) OAI221 门

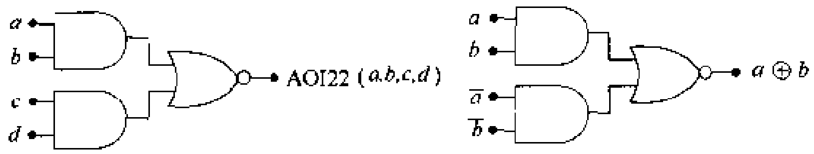
图 2.58　一般的命名方法

一般化的复合逻辑门为利用一般门来建立不同的逻辑功能提供了统一的基础。作为一个简单的例子，考虑图 2.59(a)中的 AOI22 门，它给出的输出为

$$\text{AOI22}(a, b, c, d) = \overline{a \cdot b + c \cdot d} \tag{2.75}$$

为了建立一个异或电路,可以如图 2.59(b)那样定义输入,即可以写出

$$a \oplus b = \mathrm{AOI22}(a, b, \bar{a}, \bar{b}) \tag{2.76}$$



(a) 一般的 AOI22 门　　(b) XOR

图 2.59　用一般门实现 AOI22 门的例子

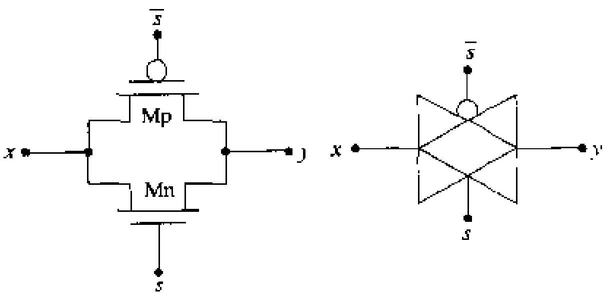
同理,异或非功能可通过下式获得

$$\overline{a \oplus b} = \mathrm{AOI22}(a, \bar{b}, \bar{a}, b) \tag{2.77}$$

这表明如何使用一般逻辑门设计随机逻辑。

2.5　传输门(TG)电路

CMOS **传输门**是通过将一个 nFET 和 pFET 并联起来构成的,如图 2.60 所示。其中 nFET Mn 由信号 s 控制,而 pFET Mp 由反信号 $\bar{s}$ 控制。当以这种方式连接时,这一对管子的作用就像是输入和输出变量分别为 x 和 y 之间的一个性能好的开关。



(a) 晶体管电路　　(b) 符号

图 2.60　传输门(TG)

这个开关的作用可以通过 s 的两种情况来理解。若 $s=0$,则 nFET 截止;因为 $\bar{s}=1$,所以 pFET 也截止,于是传输门(TG)的作用像一个断开的开关。这时在 x 和 y 之间没有关系。相反,当 $s=1$,$\bar{s}=0$ 时,两个 FET 都导通,传输门在 x 和 y 之间提供一个良好的导通路径。逻辑上,这与一个 nFET 的开关状况等同,所以可以写为

$$y = x \cdot s \quad \text{iff} \quad s = 1 \tag{2.78}$$

这里设 x 为输入,y 为输出。然而,TG 属于**双向**开关类型。图 2.60(b)中的 TG 符号就出自于这一考虑。它采用两个背对背的箭头表明数据可以沿任一方向流动。控制由 s 和 $\bar{s}$ 完成,小圈表明与 pFET 的栅连接。

传输门非常有用,因为它们可以由左至右(或反向)传送全范围的电压 $[0, V_{DD}]$。这是由

于晶体管的并联连接，零电平由 nFET 传送，而 pFET 则负责传送电源电压 V_{DD}。在现代 VLSI 中使用传输门的主要缺点在于它们要求两个 FET 和一个没有明说的反相器以便将 s 转变为 $\bar{s}$。

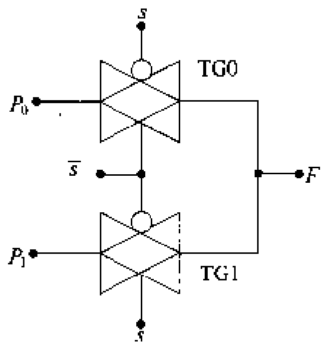
逻辑设计

多年来，在 CMOS 设计中广泛采用传输门的逻辑设计。开关操作非常简单且能传送全范围的电压，这对许多应用来说是十分有吸引力的。在许多 ASIC 结构中都可找到传输门，因此值得对它进行细致的研究。

1. 多路选择器(MUX)

传输门的理想开关特性在构造某些独特的电路时十分有用。图 2.61 的 2:1 MUX 就是一个例子。该电路的工作情况概括于表中。当选择信号 $s=0$ 时，TG0 闭合而 TG1 断开，所以 P_0 被传送至输出。如果 $s=1$，情况正好相反，TG0 断开而 TG1 闭合，这时 $F=P_1$。将这些结果合在一起得到

$$F = P_0 \cdot \bar{s} + P_1 \cdot s \tag{2.79}$$



S	TG0	TG1	F
0	闭合	开路	P_0
1	开路	闭合	P_1

图 2.61　基于 TG 的 2:1 多路选择器

上式即为所要求的方程。注意，使用“TG 对”避免了输出浮空(断路)的可能性，因为在一个传输管断开时另一个传输管总会闭合。这个 2:1 的结构可以采用二位选择字($s_1\ s_0$)扩展为 4:1 的 MUX，其中($s_1\ s_0$)的值为(0 0)，(0 1)，(1 0)和(1 1)。每条输入线(P_0, P_1, P_2, P_3)路径上有两个 TG，因此其输出为

$$F = P_0 \cdot \overline{s_1} \cdot \overline{s_0} + P_1 \cdot \overline{s_1} \cdot s_0 + P_2 \cdot s_1 \cdot \overline{s_0} + P_3 \cdot s_1 \cdot s_0 \tag{2.80}$$

例如，在 P_0 路径上的两个 TG 在($s_1\ s_0$)=(0 0)时闭合。它的电路设计留给读者作为练习。

2:1 MUX 经过修改，可以产生其他有用的功能。图 2.62(a)是其中之一。图上部 TG 的输入为 a，反相后的 $\bar{a}$ 进入下面的 TG。变量 b 和它的反信号用于控制 TG。当 $b=0$，上面的 TG 闭合，将 a 送到输出；而 $b=1$ 时则闭合下面的 TG，将 $\bar{a}$ 送到输出，于是得到

$$a \cdot \bar{b} + \bar{a} \cdot b = a \oplus b \tag{2.81}$$

即该电路实现了异或(XOR)功能。该表达式可用 2:1 MUX 的结果来验证。如果将 b 和 $\bar{b}$ 互换，即可得到一个异或非(XNOR)功能

$$\overline{a \oplus b} = a \cdot b + \bar{a} \cdot \bar{b} \tag{2.82}$$

经过这一简单变动所得到的电路见图 2.62(b)。

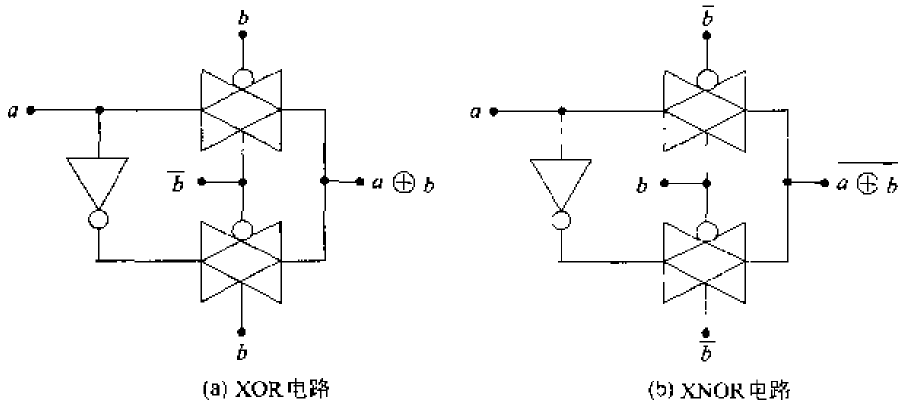


图 2.62　基于 TG 的异或电路与异或非电路

2. 或门(OR 门)

传输门的特点可以用来构建一个简单的或门电路,如图 2.63 所示。这是很有用的,因为互补 CMOS 门只能提供或非操作。电路的工作情况可以通过考察 a 对开关的影响来理解。如果 $a=0$,pFET 截止(因为 $\bar{a}=1$ 使其截止),而 TG 的作用像一个闭合的开关,于是输出 $f=b$。如果 $a=1$,则 pFET 断开,$f=a=1$ 即 $a=1$ 被传送到输出。因此,无论哪个输入为 1,输出均为 $f=1$,这就实现了或操作。还可以用逻辑方程表示 TG 和 pFET,写成

$$\begin{aligned} f &= a \cdot (\overline{\bar{a}}) + \bar{a} \cdot b \\ &= a + \bar{a} \cdot b \\ &= a + b \end{aligned} \tag{2.83}$$

以上,最后一步由逻辑吸收规则得到,这验证了比较简单的逐位分析方法。

3. 另一种异或/异或非电路

像在或门电路中一样,把 TG 和 FET 混合起来可以实现许多不同的基本逻辑门设计。由于异或(XOR)及异或非(XNOR)功能在加法器和检错/纠错算法中的重要性,这些设计中许多都是围绕它们进行的。

这一类型电路的一个例子是图 2.64 中的异或非电路。它用输入对$(b,\bar{b})$控制传输门。为了理解该电路的操作,注意到异或非门的输出在且仅在两个输入相等时才为 1。设 $b=1$,则 TG 作用同一个闭合的开关,a 被传送至输出,使 $g=a$。在这种情况下,输出在且仅在 $a=1$ 时为 1。如果 $b=0$,电路的操作则不相同。此时 TG 截止,a 直接通到 Mp/Mn 这对管子的栅上。由于 $b=0$ 连到 nFET Mn 的源,$\bar{b}=1$ 连到 pFET 的源(上面一端),$(b,\bar{b})=(0,V_{DD})$这对输入为 FET 提供电源,结果形成一个反相器! 在这种情况下,输出 $g=\bar{a}$,所以在且仅在 $a=0$时 g 为 1。这就使电路如上面所说,成为一个异或非门。将 b 和 $\bar{b}$ 互换则得到或非门。

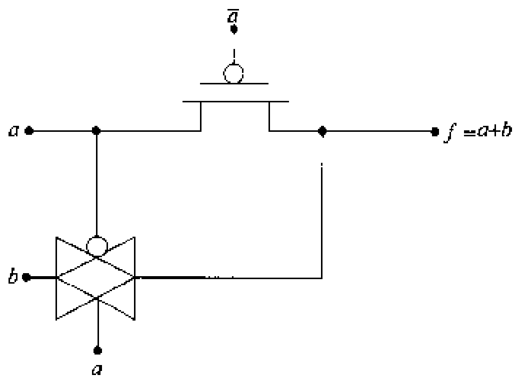


图 2.63　基于 TG 的 OR 门

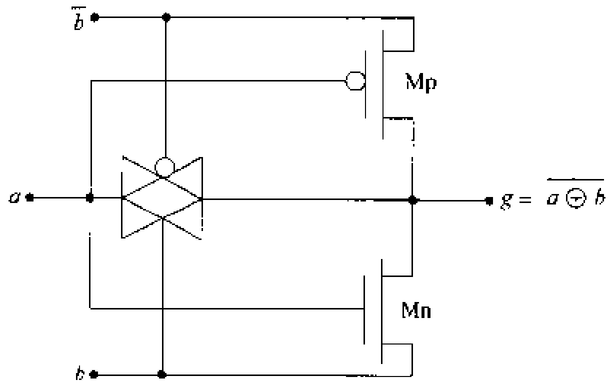


图 2.64　同时采用 TG 和 FET 的 XNOR 门

2.6　时钟控制和数据流控制

同步数字设计依赖于用时钟信号 ϕ 控制数据流的能力。TG 的开关特性提供一个简单的方法实现系统时钟。由于 TG 的开关要求有一对互补信号，所以在这类设计中使用了 ϕ 和 $\bar{\phi}$，其波形见图 2.65。周期 T 是一个全周期所需要的时间，单位为秒[s]，频率 f 定义为

$$f = \frac{1}{T} \tag{2.84}$$

其单位为赫兹[Hz] = [1/s]，1 赫兹意为在 1 s 内完成一个循环。假设半个周期对应的时钟值为逻辑 1，余下的另半个周期为逻辑 0。

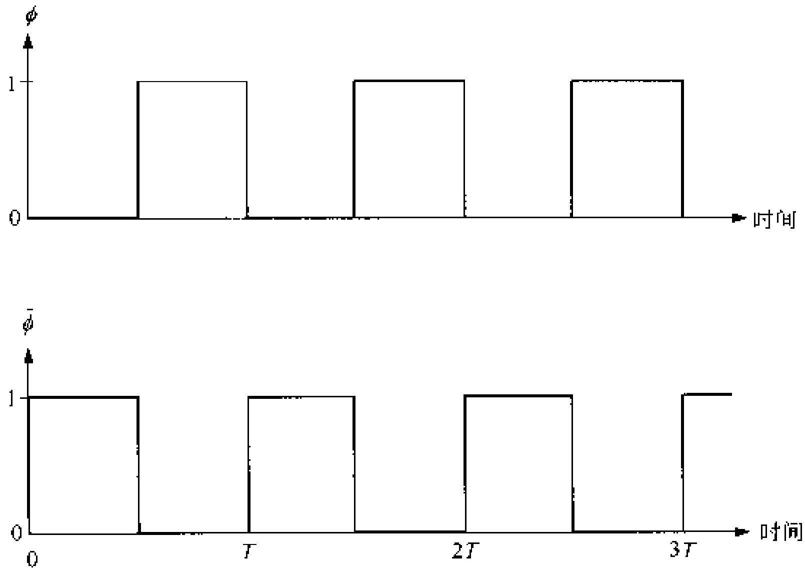
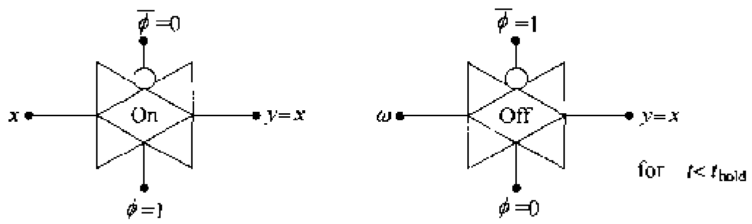


图 2.65　互补时钟信号

让我们考察一下在传输门上应用互补时钟时产生的影响。图 2.66(a)显示当时钟值 $\phi = 1$ 加到 nFET，而 $\bar{\phi} = 0$ 加到 pFET 时，TG 导通，其作用如一个闭合的开关。若将这两个值交换，则使开关断开，如图 2.66(b)所示。在静态情况下，当开关断开时，y 的值不得而知。但是 CMOS 的电特性使得在一个非常短的时间 t_{hold} 内暂时保持住 $y = x$ 的值；一般 t_{hold} 小于 1 s。如果采用高频时钟，则每半个时钟周期就会发生一次周期性的开关变化。只要$(T/2) < t_{hold}$，

输出节点就可保持原来的值。这就为在复杂电路中控制数据流提供了精确的时间基础。



(a) 开关闭合　　(b) 开关开路

图 2.66　时钟控制 TG 的工作情形

为了使用时钟控制的 TG 控制数据流,在逻辑模块的输入和输出上放上反相的 TG。图 2.67 是一个门级电路的例子。当时钟为高 $\phi=1$ 时允许左边的输入进入;第一组逻辑门对输入位求值,并且在此期间产生输出 f 和 g。因为输出 TG 截止,上述输出将保持到时钟变为 $\phi=0$。这时,f 和 g 就可进入下一组逻辑门并产生 F, G 和 H。它们将在输出端保持,直到时钟回到 $\phi=1$ 值,它们被传送出去时为止。这就是通过系统的数据流是如何用时钟控制 TG 来同步的。

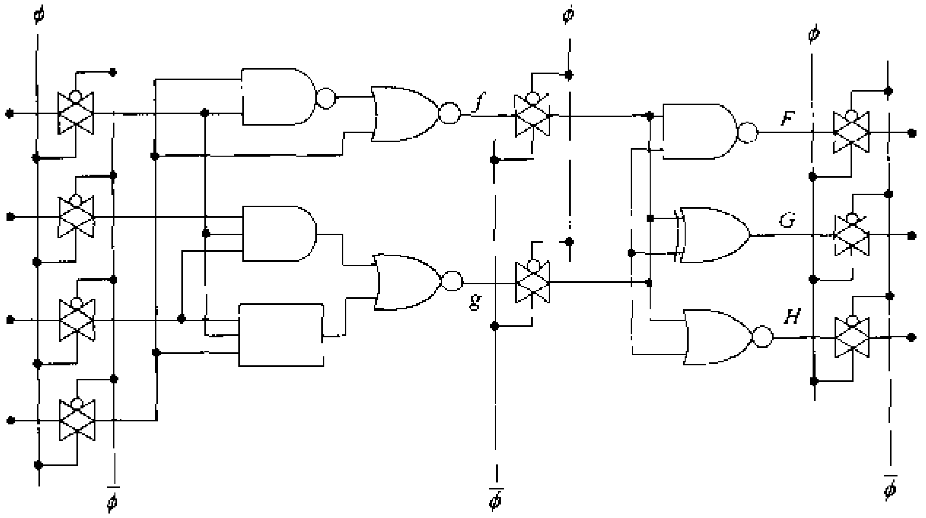


图 2.67　用传输门实现数据同步

通过系统级模块的时序图,可以看到如图 2.68 的数据流。图中每个时钟平面用依次相邻的 ϕ 和 $\bar{\phi}$ 虚线画出。它们代表在每个输入上的一个时钟控制 TG。当时钟变量为高(等于 1)时,数据可从平面的一边通向另一边。否则数据将维持在左侧直到发生时钟转换。按照图中的标记,这就是说,当时钟 $\phi=1$ 时,允许输入进入逻辑模块 1。当时钟变为 $\phi=0$ 时,前面的输出传到逻辑模块 2,如此继续下去。采用这一技术,数据每半个周期通过一个逻辑模块。由于逻辑模块可以是任意的,所以这一技术可以作为构成非常复杂逻辑链的基础。这也能将一个 n 位二进制字的每位操作保持同步。

图 2.69(a)是一个同步的字加法器。输入字 $a_{n-1}\cdots a_0$ 和 $b_{n-1}\cdots b_0$ 由 ϕ 时钟平面控制,它们的和 $s_{n-1}\cdots s_0$ 则在 $\phi=0$ 时传至输出。字中的每位在同一时间从一点传到另一点,可以跟踪通过系统的数据流。这个方法可以延伸到很大的规模,图 2.69(b)中的 ALU(算术逻辑单

元)就是一个例子。输入 A 和 B 在 ϕ 平面的控制下,由门选通进入 ALU;结果在 $\overline{\phi}=1$,也就是 $\phi=0$ 时传送到下一级。这表明在 VLSI 设计中,用时钟控制数据传输的威力。

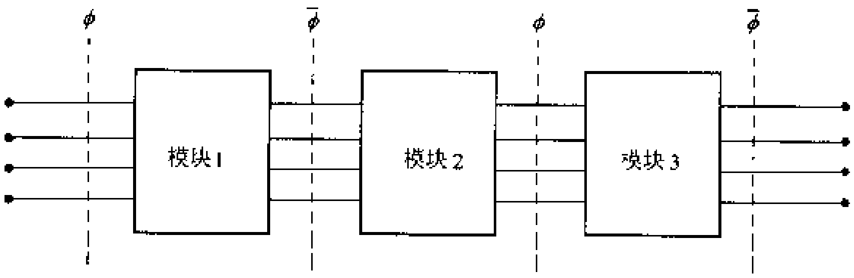
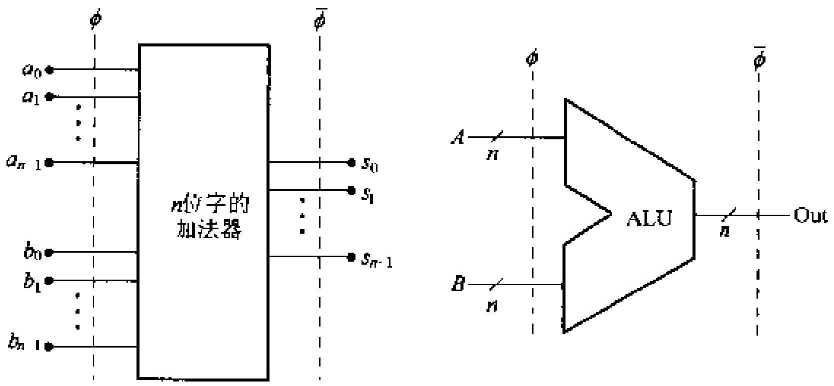


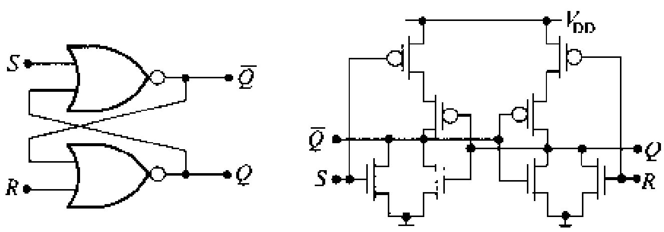
图 2.68　模块级系统时序图



(a) 钟控加法器　　(b) 钟控 ALC

图 2.69　用时钟平面控制二进制字

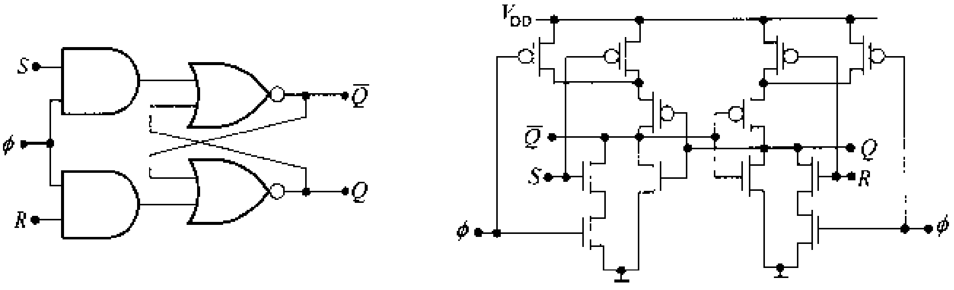
时钟控制传输门使信号流动同步,但连线本身保存数值的时间不能超过 t_{hold},这是一个非常短的时间,因此需要一个像锁存器这样的存储元件来长期保存数据位。图 2.70(a)是一个以或非门为基础的简单 SR 寄存器的逻辑图。图 2.70(b)中的 CMOS 电路是通过将两个 NOR2 门连接在一起构成的。



(a) 逻辑图　　(b) CMOS 电路

图 2.70　SR 锁存器

时钟控制可以通过与门输入端加到电路中,得到如图 2.71(a)更改后的逻辑图。它只允许在 $\phi=1$ 时改变输入。注意到用两个相同的 CMOS AOI 电路就可构成一个如图 2.71(b)所示的简单 CMOS 电路。从逻辑图起始来设计 CMOS 电路已成为一条捷径,这使 CMOS 比较容易采用。具有挑战意义的是如何使电路尽可能的快和尽可能的简单。



(a) 逻辑图　　(b) CMOS 电路

图 2.71　时钟控制 SR 锁存器

2.7 参考资料

[1] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2000.

[2] Michael John Sebastian Smith, **Application Specific Integrated Circuits**, Addison-Wesley, Reading, MA, 1997.

[3] John P. Uyemura, **A First Course in Digital Systems Design**, Brooks-Cole Publishers, Monterey, CA, 2000.

[4] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[5] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.

[6] Neil H. E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, Reading, MA, 1993.

[7] Wayne Wolf, **Modern VLSI Design**, 2nd ed., Prentice-Hall PTR, Upper Saddle River, NJ, 1998.

2.8 习题

[**2.1**] 设 $V_{DD}=5$ V, $V_{Tn}=0.7$ V。对以下输入电压值分别求出图 P2.1 中 nFET 的输出电压 V_{out}：(a) $V_{in}=2$ V；(b) $V_{in}=45$ V；(c) $V_{in}=3.5$ V；(d) $V_{in}=0.7$ V

[**2.2**] 考虑图 P2.2 中两个 pFET 的串联。电源电压为 $V_{DD}=3.3$ V，nFET 的阈值电压为 $V_{Tn}=0.55$ V，求 V_{in}在下列值时串联链右边的输出电压 V_{DD}：(a) $V_{in}=2.9$ V；(b) $V_{in}=3.0$ V；(c) $V_{in}=1.4$ V；(d) $V_{in}=3.1$ V

[**2.3**] 如图 P2.3 所示，一个 nFET 的输出用来驱动另一个 nFET 的栅。设 $V_{DD}=3.3$ V，$V_{Tn}=0.6$ V，求当输入电压为下列各值时的输出电压 V_{out}：(a) $V_a=3.3$ V，$V_b=3.3$ V；(b) $V_a=0.5$ V，$V_b=3.0$ V；(c) $V_a=2.0$ V，$V_b=2.5$ V；(d) $V_a=3.3$ V，$V_b=1.8$ V

[**2.4**] 用一个 8:1 MUX 设计一个三输入与非门(NAND3)。

[**2.5**] 以 8:1 MUX 为基础，设计一个三输入或非门(NOR3)。

[**2.6**] 考虑两输入异或(XOR)功能 $a \oplus b$

(a) 用一个 4:1MUX 设计一个异或门(XOR)。

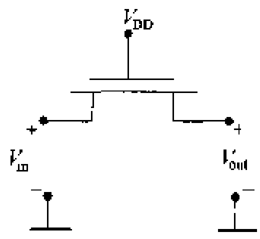


图 P2.1

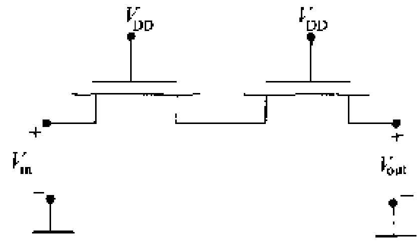


图 P2.2

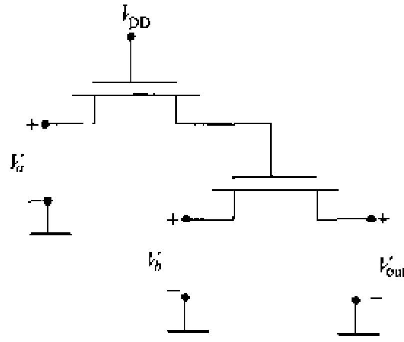


图 P2.3

(b) 修改(a)中的电路得到一个两输入的异或非门(XNOR)。

(c) 一个全加法器的输入为 a,b 和 c 并计算出和位

$$s = a \oplus b \oplus c \tag{2.85}$$

用你设计的以 MUX 为基础的门,设计一个具有该输出的电路。

[**2.7**] 用最少数量的晶体管设计一个具有下列功能的 CMOS 逻辑门

$$f = \overline{a \cdot b + a \cdot c + b \cdot d} \tag{2.86}$$

[**2.8**] 用最少数量的晶体管为以下 OAI 表达式设计一个 CMOS 电路

$$h = \overline{(a+b) \cdot (a+c) \cdot (b+d)} \tag{2.87}$$

[**2.9**] 构建一个具有下列功能的 CMOS 逻辑门

$$g = \overline{x \cdot (y+z) + y} \tag{2.88}$$

从最少数量晶体管的 nFET 电路开始,然后利用小圈推移法找出 pFET 的连线方式。

[**2.10**] 利用串-并联逻辑,设计一个完成下列功能的 CMOS 逻辑门

$$F = \overline{a + b \cdot c + a \cdot b \cdot c} \tag{2.89}$$

目标是使晶体管的数目最少。

[**2.11**] 考虑图 P2.4 表示的逻辑关系,为实现 F 设计单个的 CMOS 复合逻辑门。

(a) 用逻辑图构建 nFET 阵列。

(b) 利用小圈推移法得到 pFET 逻辑,然后按规则构成 pFET 阵列。

[**2.12**] 图 P2.5 为一 AOAI 逻辑门

(a) 利用逻辑图构建 nFET 阵列。

(b) 利用小圈推移法得到 pFET 逻辑,根据 pFET 规则按该图构建 pFET 阵列。

[**2.13**] 图 2.6 为一 pFET 逻辑阵列,利用 pFET 逻辑方程画出逻辑图。然后构建 nFET 电路。

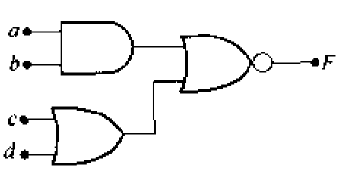


图 P2.4

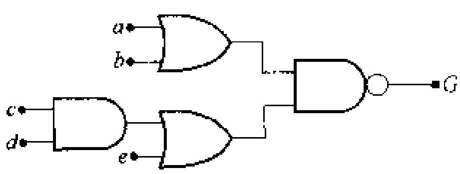


图 P2.5

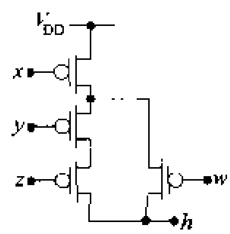


图 P2.6

[**2.14**] 用 TG 开关设计能够实现方程(2.80)功能的 4:1 多路选择器电路。

[**2.15**] 用一个 AOI22 门设计一个 2:1 MUX。设计中允许用反相器。

[**2.16**] 用三个 2:1 TG 多路选择器设计一个 4:1 MUX。

[**2.17**] 已知一个 CPU 时钟 ϕ 的频率为 2.1 GHz,求周期 T。

[**2.18**] 设 TG 的维持时间为 $t_{hold} = 120$ ms。采用如图 2.67 所示的技术,能够用来控制数据流的最低时钟频率是多少?

第 3 章　CMOS 集成电路的物理结构

CMOS 集成电路是用一组复杂的物理和化学过程，在硅圆片上的一个小面积上形成的电子开关电路。VLSI 设计者的一项基本任务是将电路图转化为硅的形式。这一过程称为**物理设计**，它是区分 VLSI 领域与一般数字工程的一个方面。本章将在层次化设计中微观硅片层次上考察 CMOS 集成电路的结构。

3.1　集成电路工艺层

一个硅集成电路可以看成是一组形成图形的材料层的集合。这些材料层具有各自的导电特性，它们可以是具有良好导电性的**金属**，也可以是阻断电流的**绝缘体**。还有一种材料层是**硅**元素，它属于**半导体类**，意即它是"部分"导体。有时把金属和硅都看做导体，但一定要弄清二者间的差别。

一个集成电路是由按特定次序将不同材料层叠在一起形成三维结构构成的，这些结构共同作用作为一个电子开关电路。每层都按系统设计工艺的规定具有预先确定的图形。这一点可以通过图 3.1 来了解。图中画出两个不同的材料层。下面是在称为"衬底"的基础材料上的薄层绝缘体。上面是形成图形的金属材料层，标记为"M1 层"。图形由两条平行**线**材料构成，放在绝缘体的上面，其位置用虚线表示。图 3.2 所示是叠放完成后的结构。图 3.2(a) 为从末端看的侧视图，可以看到这两条 M1 层的线位于绝缘层的上面。图 3.2(b) 是顶视图，可以看到这两条线相平行。图中明确画出了绝缘层，但经常在顶视图中将其省略，但仍认为它是存在的。实际上这是可接受的，因为绝缘体本身通常是一层**二氧化硅**(SiO_2)，一般称为**石英玻璃**，是透明而看不见的。这个例子虽然简单，却体现了硅集成电路(IC)分层的一个主要特点：形成图形的导体层在玻璃绝缘体的上面。复杂的 VLSI 芯片采用几层这类结构的铝或铜的导体层。

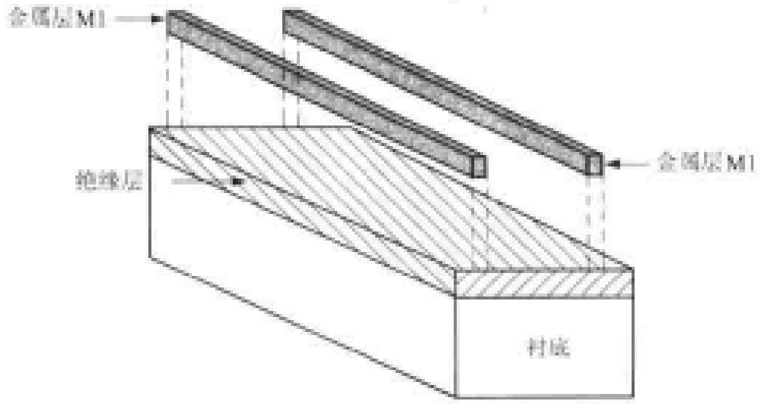


图 3.1　两种不同的材料层

上图介绍的概念可通过增加更多层数而加以扩展。假设要在图 3.2 的结构上再放一层金属图形，首先要在表面再盖上一层绝缘玻璃以避免与 M1 接触，然后使它经过**化学－机械平面化**(CMP)工序。在 CMP 中，将其表面刻蚀和"磨平"，为下一层提供一个平坦的表面。接着，在它的上面盖上第二层金属(M2 层)，得到图 3.3 所示的结构。图 3.3(a)所示的侧视图显示了所加的绝缘体覆盖了 M1 层，以及在绝缘体上面的金属层 M2。它表明了各层叠放的顺序，却并未显示两个金属层有不同的图形。图 3.3(b)所示的顶视图提供了图形的各自特点。具体来说，M2 的形状是一单根金属线，它与 M1 层的平行线垂直。从图中可以看到，当二者相交时，该线覆盖在 M1 层图形的上面。还要注意，顶视图虽未明确画出绝缘层，但要记住，二者相互并不接触。

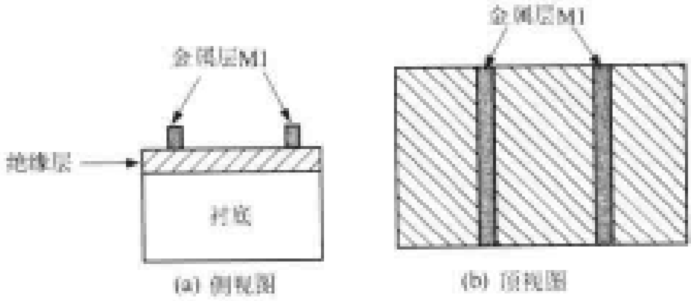


图 3.2　叠放过程完成后的各层情形

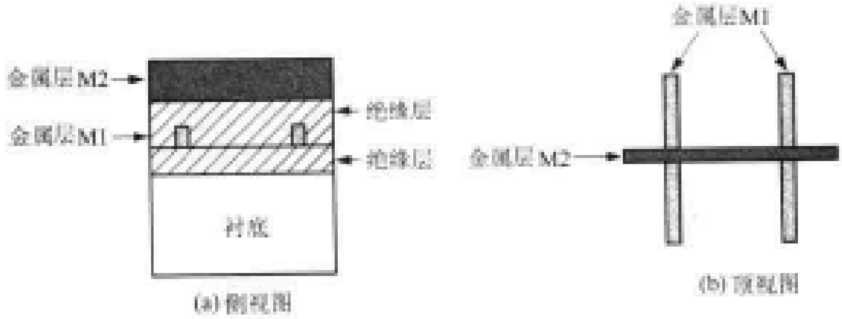


图 3.3　加上另一层绝缘层和第二层金属层

将集成电路的顶视图和侧视图结合在一起，可以看到它的三维结构了。从这个例子中可以说明下面几个重要的特点：

- 侧视图显示叠放顺序
- 绝缘层将两金属层分隔开，所以它们在电气上不同
- 每层的图形由顶视图表示

叠放顺序是在制造过程中完成的，VLSI 设计者不能改变它。但是，建立每层图形是芯片设计过程中的关键部分，因为它确定了所有 MOSFET 的位置和尺寸，并且规定如何将晶体管连接在一起。

互连线的电阻和电容

逻辑门之间通过点与点之间的信号流路径相互通信。在集成电路层次上，这一任务是通过形成图形的金属线作为导线来传导电流而完成的。这些线通常称为**互连线**。这看上去似乎只是简单的传送，其实电流的大小还取决于这些金属线材料的物理性质及其尺寸。这就意味着信号的传递速度直接受到布线物理实现的影响，这是芯片设计中的一个很重要的方面。

给形成图形的金属线加上电压 V（单位为 V），产生电流 I（单位为 A）。根据欧姆定律，对于一个像金属这样的简单导体，其电压与电流的关系为

$$V = IR \tag{3.1}$$

式中 R 为比例常数，称为**电阻**。电阻的单位为**欧姆**，并用希腊大写字母 Ω 表示。它的基本单位为伏特/安培。欧姆定律只适用于在电子学中称为**电阻器**的简单器件。电阻器的符号见图 3.4。锯齿形的线表明该器件对电流有阻碍作用。该符号只限用于电压与电流成正比的“线性”电阻器。

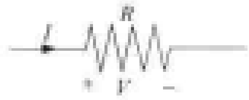


图 3.4　线性电阻符号

现在回想一下在 CMOS 电路中，布尔变量 x 由电压 V_x 代表。当该电压加在形成图形的金属线上时会产生电流 I_x。I_x 的实际值取决于该**线电阻** R_{line} 和其他一些电参数。R_{line} 的大小用单位欧姆来度量，并被认为属于不可避免的寄生（不希望有的）电气元件。电阻阻碍了电信号的流动，所以应保持 R_{line} 的值越小越好。

对于一个给定的金属线，其电阻 R_{line} 的值可用图 3.5 中的几何图形来计算。线长用符号 l 表示，其单位为厘米(cm)。截面积 A（单位 cm^2）是宽度 w 和层厚 t 的乘积

$$A = wt \tag{3.2}$$

图中的**电导率** σ 是该层所用材料的特性。其度量单位为 $[\Omega\cdot cm]^{-1}$，表示电流容易通过的程度：σ 值越大，材料层的导电性就越好。金属的电导率大，而绝缘体的电导率则很小。假设 σ 的数值是已知的。有了这些参数，线电阻的欧姆数就可由下式计算得到

$$R_{line} = \frac{l}{\sigma A} \tag{3.3}$$

上式表明了 R_{line} 的一个重要关系，即它的大小与线的长度 l 成正比，与其截面积 A 成反比。

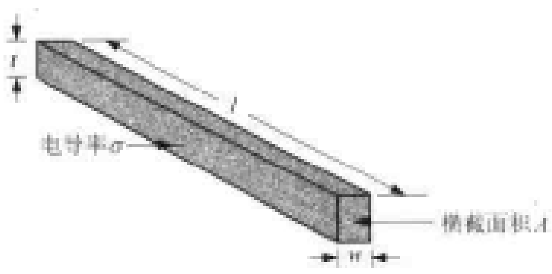


图 3.5　导线的几何图形

电阻率 ρ 是电导率的倒数，即

$$\rho = \frac{1}{\sigma} \tag{3.4}$$

ρ 的单位为[$\Omega\cdot$cm]。电阻率高意味着电导率低。采用简单的替代,线电阻公式可变为

$$R_{\text{line}} = \rho\frac{l}{A} \tag{3.5}$$

一个 VLSI 设计者不能控制 t 和 σ 的值,因为这些是在生产过程中确立的。正因为如此,将上式写成下面的形式将会非常有用

$$R_{\text{line}} = \left(\frac{1}{\sigma t}\right)\left(\frac{l}{w}\right) \tag{3.6}$$

这就把与工艺有关的各项组合到了一起。用它可以定义线的薄层电阻 R_s 为:

$$R_s = \frac{1}{\sigma t} = \frac{\rho}{t} \tag{3.7}$$

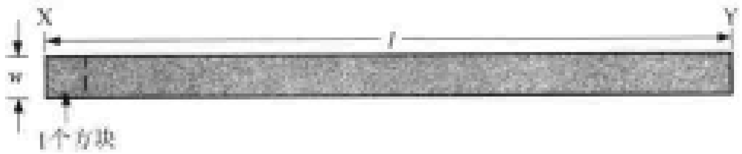
很容易验证 R_s 的单位为 Ω。材料层的薄层电阻非常有用,这是由于两个原因。第一,它可以在实验室中直接测定,而不需要知道 σ 或 t 的实际值。第二,当线的长度等于宽度($l = w$)时,其线电阻为:

$$R_{\text{line}} = R_s\left(\frac{w}{w}\right) = R_s \tag{3.8}$$

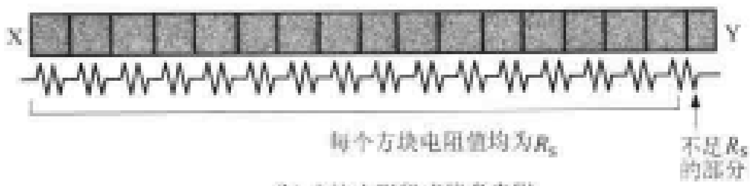
换句话说,R_s 代表一个上表面边长为($w \times w$)的方形区域的电阻。所以常用"每方 Ω"为单位表示 R_s。对薄层电阻的这种解释可以用来推导一种计算线电阻值的简便方法。

考虑图 3.6(a)的顶视图,图中已标出了一个方块。根据定义,该方块从一端到另一端的电阻为 R_s。将许多这样的方块连成一线,可以组成相当于如图 3.6(b)的一条导线。要计算该线从一端到另一端的电阻 R_{line},我们注意到每个小方块的电阻为 R_s,所以一串电阻串联等同于一个电阻器,其电阻值等于每个方块电阻的和。如果总共有 n 个方块,则

$$R_{\text{line}} = R_s n \tag{3.9}$$



(a) 视图



(b) 方块电阻组成线条电阻

图 3.6 一条图形线的顶视图

式中

$$n = \frac{l}{w} \tag{3.10}$$

是整条线从一端至另一端含有的方块数。注意 n 不必一定是整数值,也可以是分数,如图中

线的右端所示。

这一分析表明，对于一个给定的工艺层，它的线电阻取决于形成图形的线的长宽比(l/w)。这一结论的重要性是基于对信号沿线的传递速度受 R_{line} 值影响的定性观察。R_{line} 值小，允许通过的电流大，这符合高速设计的需要。以后还将对这些结论进行定量分析。

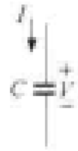


图 3.7　电容器的电路符号

互连线还表现出具有**电容**的特性，电容是存储电荷和能量的能力。电子学中将存储电荷的元件称为**电容器**，其电路符号见图 3.7。它的特点表现为其电容值 C，器件正极上的电荷 Q 为：

$$Q = CV \tag{3.11}$$

式中 V 为电压；正电荷为另一极板上的负电荷 $-Q$ 所平衡。电容的单位为**法拉**[F]，1 法拉定义为 1 库仑/伏特。由于电流定义为电荷对时间的导数 $I=(\mathrm{d}Q/\mathrm{d}t)$，求微分可得到该器件的 I-V 方程：

$$I = C\frac{\mathrm{d}V}{\mathrm{d}t} \tag{3.12}$$

电容存在于任意两个在电气上被分开的导体之间。对于互连线来说，连线导体是通过二氧化硅玻璃绝缘层与半导体衬底绝缘的。电容取决于互连线的几何形状。考虑图 3.8 中的结构，T_{ox} 是互连线与衬底之间氧化层的厚度，单位为厘米(cm)。根据基本物理学，连线电容可以用平行极板公式表示为：

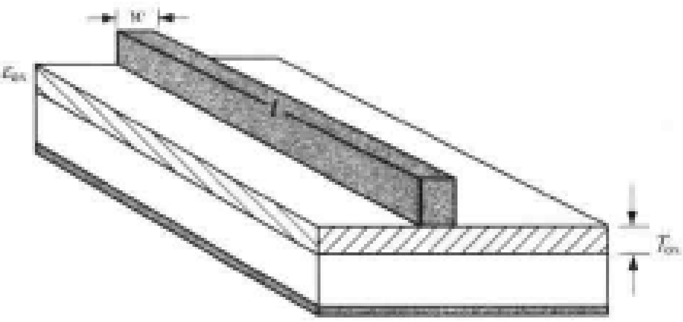


图 3.8　用于计算线电容的几何图形

$$C_{line} = \frac{\varepsilon_{ox} wl}{T_{ox}} \tag{3.13}$$

它的度量单位为法拉。在该式中 wl 为顶视图上看到的线的面积，单位为 cm^2。参数 ε_{ox} 是绝缘氧化层的介电常数，单位为 F/cm；ε_{ox} 的大小取决于氧化物的成分。

电容问题将在本章后面做更详细的研究。目前，了解互连线同时具有寄生电阻 $R_{line}[\Omega]$ 和电容 C_{line}[F]就足够了。将二者相乘得到

$$\tau = R_{line}C_{line} \tag{3.14}$$

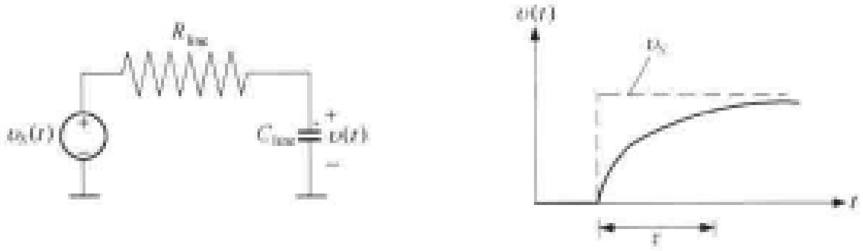
式中 τ 的单位为秒[s]，称为**时间常数**。在高速数字电路中，互连线上的信号会被延迟 τ 的时间，它是电路速度的限制因素。图 3.9 说明了这一情况。在图 3.9(a)的物理板图上，来自非门的输出信号 $v_s(t)$ 由互连线引至逻辑链的下一个门。互连线末端电压标记为 $v(t)$。如图 3.9(b)所示，寄生元件 R_{line} 和 C_{line} 用来对互连线建模。在这个简单电路中，$v(t)$ 是电容器两端的电压。如果非门的输出电压 $v_s(t)$ 使电压从 0 变为 1 电平如图中波形所示，则 $v(t)$ 也会以同样

的方式上升。但是，电容电压会滞后一个时间常数 τ，其波形也不如原来的输入源陡峭。

VLSI 工艺的许多方面都是针对如何减少 R_{line} 和 C_{line}，因此电路设计者面临着在互连线延迟的限制范围内如何设计出最快可能的开关电路。本节的简要讨论只是给出关于互连线信号传输特性的初步概念。与互连线延迟有关的问题在高密度 VLSI 芯片设计中是至关重要的。



(a) 物理结构



(b) 电路模型

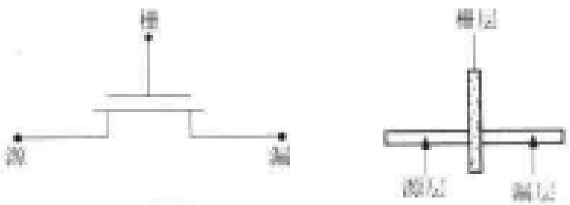
图 3.9 互连线时间常数引起的时间延迟

3.2 MOSFET

前一章的讨论侧重于用 MOSFET 开关设计逻辑门的方法。要在硅片上建立电路，首先需要了解在物理层次上 MOSFET 是什么样子，然后才能着手研究如何来设计逻辑门。

一个集成电路的 MOSFET 是由两个基本图形层的小面积组成的，它们合在一起的作用像一个控制开关。为了确定图形铺设是什么样子，可以回想一下图 3.10(a)所示的 nFET 的电路符号。这个电路符号的设计与 nFET 本身的结构很类似。每个端点提供了构成芯片上晶体管各层图形的"入口点"。这些端点已被标记为栅、源、漏，每个提供器件的入口点。由前一章的分析中知道，栅电极的作用像一个控制端口，由它接入的电压决定了开关的开闭。用电学术语来说，栅上的外加电压决定了在源端和漏端的电流流动。

现在的任务是用集成电路工艺层的概念形成一个硅场效应管。图 3.10 (b)用导体层简单表示 nFET。垂直线代表栅层，它把另外一层分为与电路图符号对应的源区和漏区。从这个简图足以了解一个集成电路晶体管的物理结构和工作情况。



(a) nFET符号　(b) nFET 工艺层

图 3.10 nFET 电路符号与相应的工艺层

通过该图可以确定物理结构中所需要的工作特性。假设将信号 G 加在栅上，我们研究 nFET 的行为。如果 $G=0$，源与漏之间没有电的连接，如图 3.11(a)所示。在该图中去掉了栅层，使该器件的行为表现得更加清楚。这时形成开路，两边在电气上被隔离；这意味着 A 和 B 之间没有关系。相反，如果应用的栅信号为 $G=1$，则 nFET 的作用像一个闭合的开关，于是源与栅之间在电气上相连，如图 3.11(b)所示。此时形成一个沟通间隙的导体层，得到如下的逻辑表达式：

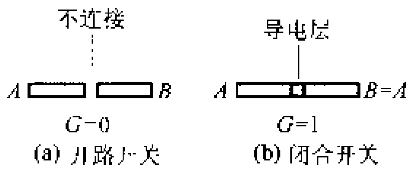


图 3.11 nFET 的简化工作图

$$B = A \tag{3.15}$$

假设漏与源在同一层上，从这一行为我们可以推导出：栅信号 G 决定了在漏区与源区之间是否存在导电区。

事实上，这就是 MOSFET 如何工作的。栅上的外加电压 V_G 用来建立一条导电通路，使电流在晶体管的漏区与源区之间流动。

我们已经看到集成电路层是如何用来构成 MOSFET 的，现在让我们更详细地考察晶体管的物理结构。图 3.12 是形成一个一般 FET 所需要的工艺层。将漏区和源区做成硅圆片中的图案，圆片相当于图 3.1 中介绍的衬底。漏区与源区虽然位于同一层，它们实际上分开一段距离 L；L 的单位为厘米[cm]，称为 FET 的**沟道长度**。漏区与源区的宽度 W 称为**沟道宽度**，也以厘米为单位。FET 的**宽长比**定义为(W/L)，它是 VLSI 设计者考虑的最重要参数。栅层与硅圆片之间用二氧化硅(玻璃)绝缘层分开。图中的垂直虚线表示叠放过程完成后各层的对准情况。

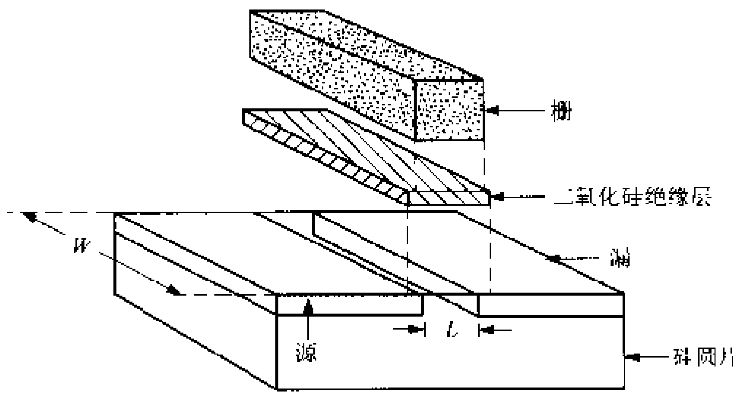
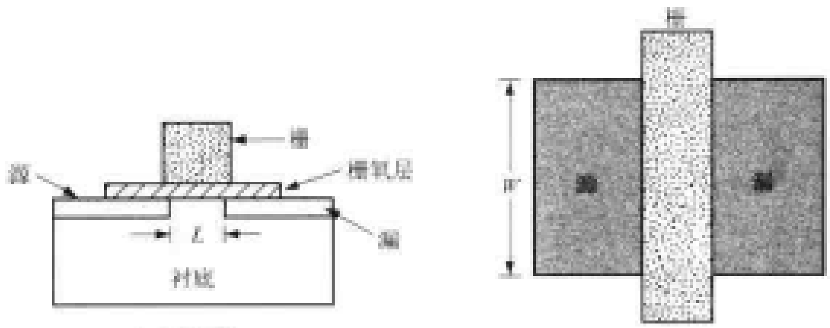


图 3.12 形成 MOSFET 的各工艺层

图 3.13 是叠放各层得到的三维结构图。3.13(a)是叠放的断面(侧视)图。这里，二氧化硅层被重新命名为**栅氧层**，这是因为它直接位于栅区的正下方。图中还清楚地标明了沟道长度 L。顶视图 3.13 (b)与在图 3.10 (b)中所画 FET 简图的形式一样。它显示了漏和源区被栅的图形分隔开。惟一的主要区别是前面的简图只画出了工艺层和导通路径，而未说明尺寸。

nFET 和 pFET 的基本结构都与图 3.13 所画的结构相同。两种器件的区别在于形成漏区与源区的工艺层的性质。它们都用硅的图形层，但是 nFET 层拥有过剩数目的带负电的电子，而 pFET 的漏－源层内具有过剩数目的正电荷。让我们简略了解一下硅物理世界，看看这

是如何形成的。



(a) 侧视图　　(b) 顶视图

图 3.13　MOSFET 视图

3.2.1　硅的导电性

纯结晶的硅是电的不良导体。它的正式名称是**半导体**，因为它能导通小电流，所以它是"不完全"的导体。硅晶体的原子密度约为 $N_{Si}=5\times10^{22}$ 个原子/cm^3，但是其中只有很少量的电子能够导电。这些电子是由于**热激发**的作用获得热能而脱离其原来的硅原子。一个纯结晶硅的样品叫做**本征**材料。每立方厘米材料中可自由载流的电子数目用符号 n_i 表示，称为**本征载流子密度**；名词"载流子"是电荷载体的简称，意为带有电荷粒子。n_i 的值是热力学温度 T 的函数。在**室温**下（$T=27℃=300K$）本征载流子的密度为：

$$n_i=1.45\times10^{10}\ cm^{-3} \tag{3.16}$$

所以晶体中只有很少一部分电子能够导电。n_i 的值随着温度的升高加大，因为有更多的热能加到这个结构上。但是与金属相比其自由电子的数量仍然很少。

如果分析纯结晶硅的化学键结构，我们发现，大多数电子被束缚在围绕原子核的轨道上。当一个电子在获得足够的热能脱离它原来的原子时，它就成为**自由**（或可动的）电子，可在晶体中移来移去。当一个电子离开它在原子中的原有位置，它就留下了一个空的共价键叫做**空穴**，见图 3.14。空穴代表缺少电子，它可看成性质与电子相反的粒子。特别是因为电子带有一个负电荷 $-q$，所以空穴就带有一个值为 $+q$ 的正电荷，① 所以它也可以加入电流移动的行列。这些粒子虽然是相互独立的，但它们在产生时则形成一个个**电子-空穴对**。

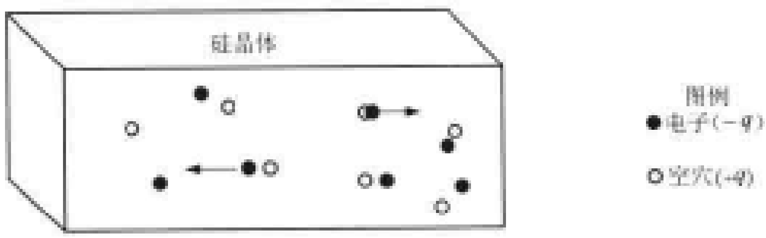


图 3.14　在硅中形成的电子一空穴对

① 基本电荷单位的数值是 $q=1.602\times10^{-19}$ 库仑。

一种材料的导电能力取决于其拥有的可自由移动的带电粒子的数目。让我们引入两个可以提供这一信息的变量。我们定义 n 为每立方厘米中自由电子的数目，p 为每立方厘米中自由空穴的数目；二者的单位均为 cm^{-3}。在一个纯净硅样品中，形成一个空穴的惟一办法是使一个电子从它的原子中释放出来。由此可见，对于我们的样品有：

$$n = p = n_i \tag{3.17}$$

将二者相乘得到

$$np = n_i^2 \tag{3.18}$$

这是**质量作用定律**的一种表达方式，它确定了在没有电流存在的情况下电子与空穴的相对数量。

纯净硅是电的不良导体，但可通过有目的地加入少量的杂质原子（称为**掺杂剂**）来改变而形成**掺杂**样品。其目的是增加电子或空穴的数量以增加其导电性。在晶体中加入砷（As）或磷（P）原子可以增加自由电子的数量。得到的样品称为 **n型**材料，因为它含有过剩的带负电荷的电子。当作为掺杂剂时，砷和磷都能为晶体提供自由电子，所以被称为**施主原子**，或简称**施主**。每立方厘米中加入施主的数量用符号 N_d 表示，其典型值的范围约为 $10^{16} \sim 10^{19}\ cm^{-3}$。每个施主原子为晶体增加一个自由电子，所以可用下式计算电子密度

$$n_n \approx N_d \ cm^{-3} \tag{3.19}$$

式中，符号 n_n 表示在一个n型样品中电子的密度。n型样品中空穴的数量用符号 p_n 表示，根据质量作用定律有

$$p_n \approx \frac{n_i^2}{N_d} \ cm^{-3} \tag{3.20}$$

在n型材料中，电子被称为**多子**，而空穴则称为**少子**，这是由于它们的相对数量不同。

【例3.1】 设施主掺杂密度为 $N_d = 2\times10^{17}\ cm^{-3}$。则电子密度为

$$n_n \approx N_d = 2\times10^{17}\ cm^{-3} \tag{3.21}$$

而空穴密度为

$$p_n \approx \frac{n_i^2}{N_d} = \frac{(1.45\times10^{10})^2}{2\times10^{17}} \tag{3.22}$$

由此可得

$$p_n \approx 1\times10^{3}\ cm^{-3} \tag{3.23}$$

显然此样品中 $n_n \gg p_n$。

与 n 型极性相反的材料叫做 p 型材料，这是通过向晶体中加入硼(B)原子形成的。p 型材料中带正电荷的空穴数多于带负电荷的电子数。加入硼是因为每个硼原子都会在化学键中形成一个自由空穴。由于一个空穴可以接受一个电子，硼被称为受主掺杂剂。每一立方厘米中加入的受主数量用符号 N_a 表示。受主的密度范围与施主的大体相仿(约为 10^{14}～10^{19} cm^{-3})，但其作用正好完全相反：加入硼会增加 p 型半导体中空穴的浓度 p_p。我们用下式计算载流子的密度

$$p_p \approx N_a \qquad n_p \approx \frac{n_i^2}{N_a} \tag{3.24}$$

同时由于 $p_p > n_p$，我们称空穴为多子载体，电子为少子载体。p_p 和 n_p 的单位均为 cm^{-3}。

一个载流子密度为 n 和 p 的半导体区域其电导率 σ 为

$$\sigma = q(\mu_n n + \mu_p p) \tag{3.25}$$

式中，μ_n 和 μ_p 分别称为电子和空穴的**迁移率**，单位为 $cm^2/(V\cdot s)$。从定性角度，迁移率是说明一个粒子"如何移动"的参数。μ 值小说明粒子移动困难，μ 值大则说明移动相对自由。室温下本征硅的迁移率为

$$\mu_n = 1360\ cm^2/(V\cdot s) \qquad \mu_p = 480\ cm^2/(V\cdot s) \tag{3.26}$$

所以它的电导率为 $\sigma \approx 4.27\times10^{-6}[\Omega\cdot cm]^{-1}$或者说电阻率为 $\rho \approx 2.43\times10^5[\Omega\cdot cm]$。为了进行比较，作为优良绝缘体的石英玻璃的电阻率 ρ 约为 $10^{12}[\Omega\cdot cm]$。

对于一个特定的 n 型样品，若 $n_n \gg p_n$，则通常可以近似其电导率为

$$\sigma \approx q\mu_n n_n \tag{3.27}$$

同样，p 型区的电导率常常可以估计为

$$\sigma \approx q\mu_p p_p \tag{3.28}$$

不过就目前的讨论，最重要的是要记住，在 n 型区以带负电荷的电子为主，而在 p 型区则大多数为带正电荷的空穴。

【例 3.2】 考虑一个硼掺杂密度为 10^{15} cm^{-3}的 p 型硅掺杂样品，其多子电荷载体为空穴，密度为

$$p_p = 10^{15}\ cm^{-3} \tag{3.29}$$

而少子载体的电子密度为

$$n_p \approx \frac{(1.45\times10^{10})^2}{10^{15}} = 2.2\times10^5\ cm^{-3} \tag{3.30}$$

已知该样品的的迁移率为 $\mu_n = 1350\ cm^2/(V\cdot s)$，$\mu_p \approx 450\ cm^2/(V\cdot s)$。其电导率为

$$\begin{aligned}\sigma &= (1.6\times10^{-19})[(1350)(2.2\times10^5)+(450)(10^{15})] \\ &= 0.072\quad [\Omega\cdot cm]^{-1}\end{aligned} \tag{3.31}$$

这相当于电阻率为

$$\rho = \frac{1}{0.08} = 13.9\ [\Omega\cdot\text{cm}] \tag{3.32}$$

简单检查一下以上各值就可以看到在该例中，$\mu_p p_p \gg \mu_n n_p$。一般来说，硅样品电阻率的数量级为$(1\sim10)\ \Omega\cdot\text{cm}$。

这一例子表明掺杂水平是决定 n 型或 p 型硅电导率的最重要因素。掺杂密度的提高能产生更多的带电粒子，这有利于导电过程。然而大量的杂质原子也造成更多粒子必须通过的势垒，从而影响它们的移动，这称为**杂质散射**，并且可以通过迁移率 μ 与总的掺杂密度 N 的关系来描述

$$\mu = \mu_1 + \frac{\mu_2 - \mu_1}{1 + \left(\dfrac{N}{N_{ref}}\right)^{\alpha}} \tag{3.33}$$

式中 μ_1, μ_2, N_{ref}和 α 为常数。对于电子来说，室温下纯硅的值大约是 $\mu_1 = 92\ \text{cm}^2/(\text{V}\cdot\text{s})$，$\mu_2 = 1380\ \text{cm}^2/(\text{V}\cdot\text{s})$，$N_{ref} = 1.3\times10^{17}\ \text{cm}^{-3}$，$\alpha = 0.91$。相应的空穴值为 $\mu_1 = 47.7\ \text{cm}^2/(\text{V}\cdot\text{s})$，$\mu_2 = 495\ \text{cm}^2/(\text{V}\cdot\text{s})$，$N_{ref} = 6.3\times10^{16}\ \text{cm}^{-3}$，$\alpha = 0.76$。迁移率随着掺杂的增加而降低，这种现象在器件物理中称为**二阶效应**。虽然在简单计算中试图忽略杂质散射，但这样做可能造成显著的误差。最后还要提到的是对于一个给定的掺杂水平 N，

$$\mu_n > \mu_p \tag{3.34}$$

这意味着电子比空穴更容易移动。从物理上来讲，只要假设电子是传统意义上的真实粒子，而空穴则是缺少"粒子"，就可以理解这一点。

以上分析假定了样品中只有施主 N_d 或只有受主 N_a 存在。但是在 CMOS 工艺中，大多数掺杂区同时存在施主和受主，其极性是由占优势的种类决定的。若要形成 n 型区，需要 $N_d > N_a$，所以施主在数量上要胜过受主。载流子的数目可用下式计算

$$n_n \approx N_d - N_a \qquad p_n \approx \frac{n_i^2}{(N_d - N_a)} \tag{3.35}$$

这时电子的数目占大多数。对于 p 型区域，我们需要 $N_a > N_d$，所以载流子密度为

$$p_p \approx N_a - N_d \qquad n_p \approx \frac{n_i^2}{(N_a - N_d)} \tag{3.36}$$

这时空穴 p_p 为多数载流子，而电子 n_p 为少数载流子。为了计算迁移率，在方程(3.33)中采用总的掺杂密度 $N = N_a + N_d$。电导率仍可用下式计算

$$\sigma = q(\mu_n n + \mu_p p) \tag{3.37}$$

因为只是值改变而已。一个特例是 $N_d = N_a$。因为施主释放的每个电子都有一个受主的空穴与之对应，该材料看上去与 $n = p = n_i$ 的本征材料一致，这称为**全补偿**。注意此时由于掺杂的数量非零，它的迁移率将小于本征值。

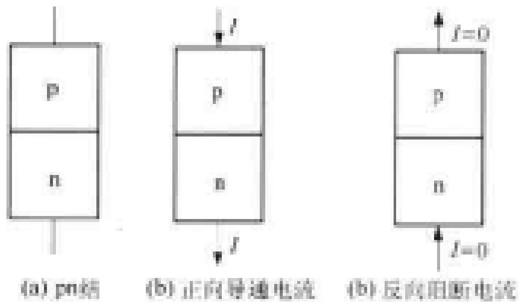


图 3.15 pn 结的形成及特性

当 n 区与 p 区彼此接触时就形成一个非常特殊的界面。这个 **pn 结**只允许沿一个方向导电,即从 p 端向 n 端导电。如果企图电流从 n 端流向 p 端,那么 pn 结就会阻止它,就像一个断开的开关一样。pn 结的特性概括于图 3.15 中。在电子学中,这一特点用来制造一种叫做二极管的器件。这种只允许电流以一个方向流动的特性叫做**整流**。

3.2.2 nFET 和 pFET

在明确 n 型区和 p 型区的差别后,可以定义 nFET 和 pFET 两种结构了。这个任务很简单:一个 FET 的极性(n 或 p)是由漏区和源区的极性决定的。器件设计成当它导电时,如图 3.11(b)所示导电层的极性与漏区和源区的极性相同。nFET 采用 n 型的漏区和源区,而 pFET 则具有 p 型的漏区和源区。这些分别显示在图 3.16(a)及(b)上。图中加上的金属接触说明如何将漏区和源区与电路的其他部分连接。

让我们首先来考察 nFET。漏区和源区都标有"n+"表明它们为**重掺杂**。这意味着施主的掺杂密度 N_d 相对较大,一般的典型值约在 $N_d = 10^{19}\ cm^{-3}$。衬底层(位于底部)现在设为 p 型,意即它有一个硼掺杂密度 N_a;受子密度的合理值可以是 $N_a = 10^{15}\ cm^{-3}$。注意,pn 结形成在 n+ 区和 p 型衬底之间。正如讨论图 3.15 时那样,它们是用来阻止电流在衬底与器件上面的 n+ 区之间流动的。

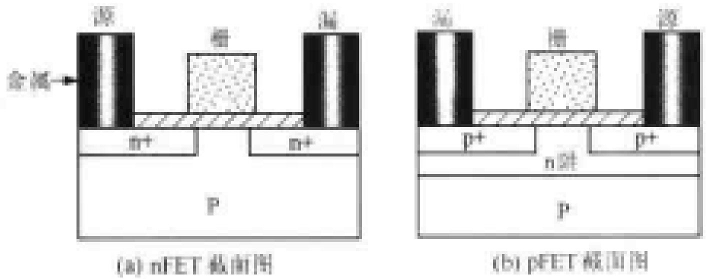


图 3.16 nFET 和 pFET 层

pFET 具有和 nFET 同样的结构,但极性相反。它的源区和漏区是埋置在 n 型"阱"层中的p+ 部分;而 n 型阱本身又居于 p 型衬底的上面。在这一器件中形成了几个 pn 结,都是用来阻止电流在相邻层之间的流动的。分层图更为复杂,因为 CMOS 设计同时用到在同一个硅圆片上形成的 nFET 和 pFET。如果选择圆片为 p 型,那么只要如图 3.16(a)所示那样加入n+ 区就可构成 nFET。如果把 pFET 的 p+ 区直接加到 p 衬底上的话,就会得不到所需要的分层结构:即 p+ 应在 n 区内。由于没有 pn 结形成,无法控制电流。为了纠正这个问题,采用了 n 阱层来构成 pFET,如图 3.16 所示。这样就保证了晶体管具有相反的电特性。

3.2.3　FET 中的电流

在 CMOS 逻辑电路中 MOSFET 用来作为电压控制开关。在栅电极上加上信号，结果就如在图 3.11 的 nFET 图上看到的，使开关开断或闭合。栅下面导电层的形成是由建在 MOSFET 本身栅区内的电容的性质引起的。图 3.17 是一个根据基本物理学构成的简单平行板电容器。这一器件由两个相同的金属板构成，它们被一个厚度为 t_{ins}（单位为 cm）的绝缘层隔开。平板面积为 A，单位为 cm^2。电容器在该两个极板上储存电荷 Q。当一个电压差 V 加在这两个极板上时，储存的电荷为

$$Q = CV \tag{3.38}$$

式中，C 为电容。对于一个平行板结构，计算电容的基本公式是大家熟悉的

$$C = \frac{\varepsilon A}{t_{ins}} \tag{3.39}$$

式中，ε 是绝缘体的介电常数，单位为 F/cm。ε 的值取决于用来隔离两块平板的材料。[2] 我们观察到的一个最重要的现象是加在上面平板上的正电压 V 在下面平板上**感应**出负电荷 $-Q$。

让我们更详细地考察一下图 3.18 所示的 nFET 管。器件的中心区设计为一个电容器。栅氧化层是位于栅（相当于上面的平板）和 p 型衬底（相当于下面的平板）之间的绝缘玻璃。在早期的 MOS 中，栅是用金属铝（Al）制造的。这一分层结构由此产生了 MOS 这个简称，即它为金属－氧化物－半导体的缩写。在现代工艺中，栅的材料是多晶体的硅，通常称为**多晶硅**，或干脆叫**多晶**。[3] 虽然栅的材料不再是金属，但 MOS 的名称从未随之改变，并仍沿用至今。[4]

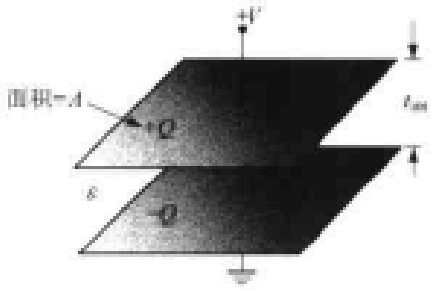


图 3.17　平行板电容器

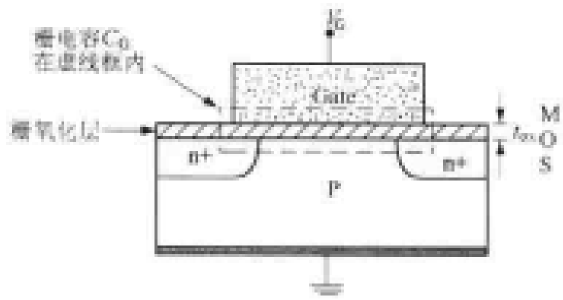


图 3.18　n 沟道 MOSFET 中的栅电容

我们引入**栅氧化层**电容来描述 MOS 结构

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{3.40}$$

单位为 F/cm^2。与平行板公式(3.39)相比它去掉了面积。其目的是使 C_{ox} 适用于电路中的任意器件。如果栅的面积为 $A_G(cm^2)$，则 FET 的**总栅电容**为：

$$C_G = C_{ox}A_G \tag{3.41}$$

② 物理学上，介电常数用来衡量材料存储电能的能力。

③ 多晶体由小区域的硅晶体(crystallites)组成；该材料将在第 4 章中详细讨论。

④ 最常用的替换词是 IGFET，代表绝缘栅 FET。

单位为法拉。公式中的 ε_{ox} 是玻璃绝缘层的介电常数。在现代工艺中，几乎所有的硅 MOSFET 都用二氧化硅，其介电常数为

$$\varepsilon_{ox} = 3.9\varepsilon_0 \tag{3.42}$$

其中 $\varepsilon_0 = 8.854\times10^{-14}$ F/cm² 是真空的介电常数。在 CMOS 中氧化层厚度 t_{ox} 是一个关键参数。最好使用较薄的氧化层（t_{ox} 较小），其原因将在后面讲到。现代生产线可以达到 $t_{ox} \leq 10$ nm = 100 Å，更先进的制造设备还具有生产比这厚度一半还小的氧化层的能力。[5]

【例 3.3】 考虑一个栅氧化层，厚度为 $t_{ox} = 50$ Å $= 50\times10^{-8}$ cm。其每单位面积的栅氧电容为

$$C_{ox} = \frac{(3.9)(8.854\times10^{-14})}{50\times10^{-8}} = 6.91\times10^{-7} \quad \text{F/cm}^2 \tag{3.43}$$

这是一个典型值。假定 FET 的栅面积为

$$A_G = (1\times10^{-4}\text{cm})\times(0.4\times10^{-4}\text{cm}) = 4\times10^{-9}\text{cm}^2 \tag{3.44}$$

注意 10^{-4} cm $= 10^{-6}$ m $= 1$ μm（微米），常称为一个 micron，是用来描述 FET 尺寸的单位。本例中的栅电容为

$$C_G = (6.91\times10^{-7})(4\times10^{-9}) = 2.76\times10^{-15} \quad \text{F} \tag{3.45}$$

定义 1 飞[母托]法拉(fF)为 1 fF $= 10^{-15}$ F，该栅电容就是

$$C_G = 2.76 \text{ fF} \tag{3.46}$$

这是一个现代器件的典型值。电子专业人员会注意到，它较日常遇到的典型电容值要小得多。

以上讨论说明 MOSFET 的栅确实是 MOS 电容器的一个极板。栅上的外加电压在电容器的另一块极板上，即在紧接栅氧化层下面的硅区上引起一个极性相反的电荷层。如果在栅上加上正电压，那么在硅区就形成负电荷层。反之，使用相对器件其他部分为负的电压，就会在硅区形成带正电荷的空穴层。硅区之所以能够形成这一电荷薄层，是因为它是一种半导体材料，其电荷载体的数量取决于所在位置的电场情况。有了这一观察结果，电流流动机理就很容易看清楚了。

首先考虑图 3.19 中的 nFET。其漏区和源区为 n 型，但它们被 p 型衬底的一部分隔开。图 3.19(a)中栅电压的值为 0 V，所以栅氧化层下面没有产生感应电荷。由右面硅片的顶视图可以看到，漏和源是隔开的，所以它们之间没有电流能够通过。这是由于 pn 结的电流阻断特性使它类似于一个断开的开关。当在栅上加上正电压如图 3.19(b)所示时，MOS 电容结构的栅氧化层下感应一层带负电荷的电子。该电子层在源与漏区之间建立起电的联系。电子形成一条“沟道”使电流通过 nFET，此时器件的作用像一个闭合的开关。电子层的形成要求栅电压大于第 2 章中介绍过的阈值电压 V_{Tn}。nFET 阈值电压的典型值为 $V_{Tn} = 0.70$ V。这一参

[5] 1 Å $= 10^{-10}$ m $= 10^{-8}$ cm

数是在生产过程中确定的,在 VLSI 设计阶段总是被假定为已知值。

沟道电荷的单位为库仑,可由下式求得

$$Q_c = -C_G(V_G - V_{Tn}) \tag{3.47}$$

式中 V_G 为栅电压,C_G 是栅电容。采用电压差($V_G - V_{Tn}$)是因为只有在 V_G 达到 V_{Tn} 时才会有电荷形成。式子前的负号表示沟道由带负电荷的电子形成。通过沟道的电流 I 可表达为

$$I = \frac{|Q_c|}{\tau_t}\ \text{C/s} \tag{3.48}$$

式中引入 τ_t 作为沟道中**渡越时间**,单位为秒。实际上,τ_t 就是电子从一个 n+ 区移动到另一个 n+ 区所需要的平均时间,可用下式进行计算

$$\tau_t = \frac{L}{v} \tag{3.49}$$

式中 v 为粒子速度,单位为 cm/s。代入电流公式(3.48)得到

$$\begin{aligned} I &= \frac{C_G}{(L/v)}(V_G - V_{Tn}) \\ &= vC_{ox}W(V_G - V_{Tn}) \end{aligned} \tag{3.50}$$

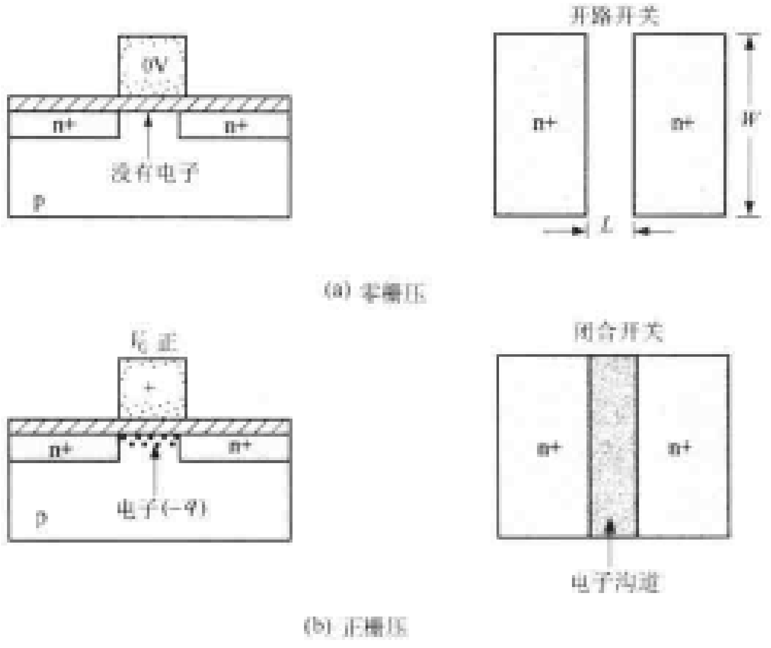


图 3.19 nFET 中对电流的控制

推导第二步时我们用了公式(3.41)中的栅电容 C_G 的定义。带电粒子在 FET 中移动的速度可以估计为

$$v = \mu_n E \tag{3.51}$$

式中,E 为电场强度,μ_n 为电子迁移率。当在两个 n+ 区之间加上一个电压 V(与栅电压无关)时,电场可近似为

$$E = \frac{V}{L} \tag{3.52}$$

单位为 V/cm。将这些关系代入公式(3.50)得到

$$I \approx \mu_n C_{ox}\left(\frac{W}{L}\right)(V_G - V_{Tn})V \tag{3.53}$$

这是得到的电流的第一个近似值。器件的线性电阻 R_n 可以通过求以下的比值计算

$$R_n = \frac{V}{I} = \frac{1}{\beta_n(V_G - V_{Tn})} \tag{3.54}$$

式中,我们已定义参数

$$\beta_n = \mu_n C_{ox}\left(\frac{W}{L}\right) \tag{3.55}$$

它称为**器件互导**(转移电导),单位为 A/V^2。[5] 采用这一模型,可把 nFET 看成一个断开或闭合的开关器件。当它断开时漏源之间无沟道存在,$R \to \infty$;当它闭合时漏-源之间存在电阻 R_n。

值得一提的细节是在 MOSFET 分析中使用的迁移率 μ_n 是在硅表面的值,因此称为**表面迁移率**。这与公式(3.33)的计算的值不同,后者是**体迁移率**,即为材料内部的值。一个简单估计为表面迁移率大约是体迁移率值的一半。在实际电路设计中则使用实验室的测定值。

更深入的分析将发现 MOSFET 本质上为**非线性器件**,即通过 FET 的电流 I 是它两端电压 V 的非线性函数。将在第 6 章对这一关系做更详细的分析。不过,作为简单模型,经常把晶体管看成是一个线性电阻,其值为

$$R_n = R_{c,n}\left(\frac{L}{W}\right) \tag{3.56}$$

其中

$$R_{c,n} = \frac{1}{\mu_n C_{ox}(V_G - V_{Tn})} \tag{3.57}$$

是电子电流沟道的等效薄层电阻。

pFET 的行为与 nFET 类似,只是所有的极性相反。其操作情况概括于图 3.20 中。如果栅电压为正[见图 3.20(a)],则栅氧下面的 n 型层上只有负电荷存在。由于漏区和源区均为 p 型,它们被 n 型区在电气上互相隔离,从而晶体管的作用像一个断开的开关。反之,若在栅上加上负电压,则如图 3.20(b)所示,栅氧下面就可形成一层带正电荷的空穴。空穴层的形成要求在最高电压的 p+ 区和栅之间的电压差必须大于 pFET 阈值电压 $|V_{Tp}|$ 的值。这时,在 p 型的源和漏区之间形成导电沟道,使电流能通过晶体管,于是 pFET 类似于一个闭合的开关。像 nFET 一样,pFET 也表现出存在一个电阻,可估计为

$$R_p = \frac{1}{\beta_p(V_G - |V_{Tp}|)} \tag{3.58}$$

按照惯例,V_{Tp} 为负数,所以用 $|V_{Tp}|$ 使公式具有与 nFET 相同的形式。式中

$$\beta_p = \mu_p C_{ox}\left(\frac{W}{L}\right) \tag{3.59}$$

---

⑤ 通常互导参数的单位为安培除以电压的平方或立方,等等。转移电阻的基本单位为伏特/安培。

是 pFET 的器件互导，μ_p 为空穴迁移率。

虽然对 nFET 和 pFET 导通特性的初步研究被高度简化了，但它确实为 VLSI 设计者提供有用的具体概念。常常一个简单的模型比一个复杂的模型更有用。

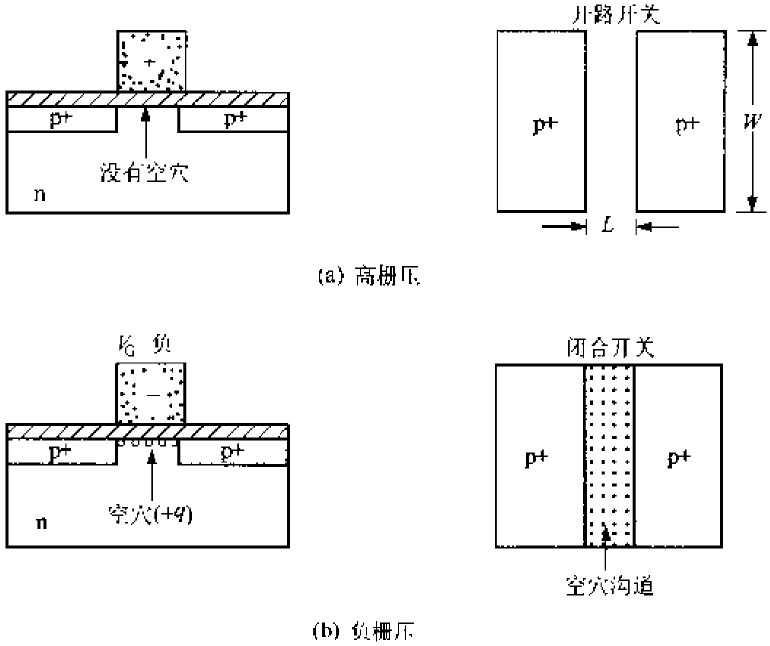


图 3.20 pFET 的开关工作情况

3.2.4 栅电容的驱动

让我们更深入了解一下 MOS 电容器系统的行为。这是 FET 工作的基础，但在 CMOS 集成电路中存在的任何电容都会引起信号延迟。图 3.21 为一个电容量为 C 的电容器电路符号。图中定义正电流 i 为流入电容具有正电压一边的电流。注意正电荷 $+Q$ 存储在上极板上，而下极板则保持有 $-Q$ 值的负电荷。流入电容器的电流 i 是时间 t 的函数，它是电荷的时间变化率。

$$i(t) = \frac{dQ}{dt} \tag{3.60}$$

由于 $Q = CV$，可以用它替代电荷，得到电容器的电流－电压(I-V)关系

$$i = C\frac{dV}{dt} \tag{3.61}$$

这告诉我们有关在 CMOS 电路中电压信号行为的几件事。第一，电容器两端的电压 $V(t)$ 不可能突变，即不能使 $dt \to 0$，否则(dV/dt)的值就会变得无穷大，但实际上不可能有无穷大的电流 i。如果把该结论应用于图 3.22 中 FET 的栅电容 C_G，可以断定，栅电压 V_G 的变化不可能没有延迟。这一延迟就是电荷移入或移出栅极所需要的时间

$$Q = C_G V_G \tag{3.62}$$

如果把它和转变 FET 通断状态的机理联系起来，即意味着晶体管本身引起信号延迟。电容

C_G 的值决定改变电压所需要的电荷量。所以电容大意味着延迟时间长。

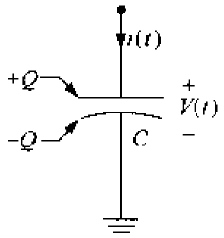


图 3.21 电容器中的电压和电流

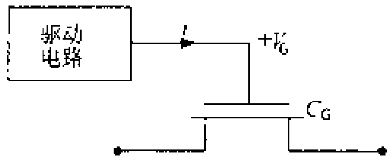


图 3.22 驱动 FET 的栅

第 2 个重要的观察是电容器储存电能。C_G 的充电和放电相当于改变器件中的储备的能量。所以，使晶体管改变通断要求在电路中把能量从一点转移到另一点。功率 P 的单位为**瓦(特)**[W]，它与能量 E 的关系为

$$P = \frac{\mathrm{d}E}{\mathrm{d}t} \tag{3.63}$$

式中 E 的单位为**焦耳**[J]。根据定义，1 瓦特功率意味着一秒钟内转移 1 焦耳的能量。对于一个电压为 V，流过的电流为 i 的电器件，功率是它们的乘积 $P = Vi$。利用电容器公式(3.61)得到

$$P = V\left(C\frac{\mathrm{d}V}{\mathrm{d}t}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(C\frac{V^2}{2}\right) \tag{3.64}$$

所以，一个电压为 V 的电容器中所储存的电能 E_c 为

$$E_c = \frac{1}{2}CV^2 \tag{3.65}$$

当应用于 CMOS 开关电路时，这意味着一个 FET 的栅电压由 0V 变为 V_{DD} 时电路中每一个这样的晶体管所需要的能量为

$$E_c = \frac{1}{2}C_G V_{DD}^2 \tag{3.66}$$

现在注意到电路驱动必须通过一个电阻为 R_{line} 的互连线来传送电流。电阻并不储存电能，相反，它们**消耗**功率并把它转变为热。由电阻 R 消耗的功率 P_R 可通过下式计算

$$P_R = Vi = i^2 R \tag{3.67}$$

这里我们运用了欧姆定律。这说明电流的流动会引起局部热效应。这一现象适用于电路中的所有电器件，而不仅限于互连线。

这些简单的观察可得出 VLSI 设计的一些关键特点，这是在全书中要研究的问题。目前直接需要考虑的两个问题是：

- 由于器件和互连线的物理特性引起的开关延迟；
- 电路中每次开关都需要能量转移，这就意味着电路中将发生功耗。

上面第一个问题意味着设计者必须了解开关延迟的本质，以便设计快速的数字电路。FET 和互连线的特点都会影响整个系统的速度，所以 VLSI 的设计要将电路作为一个整体来

考虑。第二个问题更为实际。过剩的局部发热可能很严重，以致熔化硅晶体和损坏芯片。自然这必须通过合理的设计并使用散热技术来避免。如果芯片用在以电池为电源的便携式单元中，则设计必须考虑降低功耗要求，以延长电池的寿命。

3.3　CMOS 工艺层

既然已经了解如何用形成图形的材料层来构成 nFET 和 pFET，就可上到一个更高的层次来考察 CMOS 集成电路的整体结构。

CMOS 为世界计算机工业提供了经济基础。许多公司在市场中竞争，每个都试图提供比其他公司更先进的技术基础。由于 21 世纪头几年中高密度电路生产技术的快速发展，已给 CMOS 带来无数的变化。在此，将选择学习较为简单的工艺，有目的地避开高级的（因而也是复杂的）技术。特别将集中研究 **n 阱工艺**作为典型例子。

首先定义什么是“CMOS 生产工艺”。简单来说，这是有关从裸硅“圆片”到形成最终的电子集成电路所需要的一系列步骤。生产工艺的细节将在第 5 章讨论。目前将只考虑它最终的结构。

n 阱工艺起始于 p 型衬底（圆片），它用来作为构造所有晶体管的基础层。nFET 可以直接在 p 型衬底上形成，加上 n 阱区是为了包括 pFET。图 3.23 的截面图显示已经在衬底上制造的 nFET 和 pFET 的结构。从图中可以识别下列类型的工艺层：

- p 衬底
- n 阱
- n+（nFET 漏/源）
- p+（pFET 漏/源）
- 栅氧化层
- 栅（多晶硅）

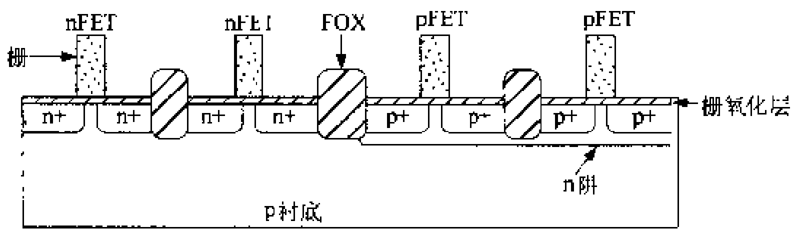


图 3.23　n 阱工艺中的 MOSFET 各层

注意，名词“工艺层”意指具有不同电特性的区域，虽然它实际上可与另一层（例如 n+ 和 p+ 区）处于同一几何层上。图中还有一个标有“FOX”的区域，它定义了**场氧区**。**场氧区**只是凹进去的绝缘玻璃（二氧化硅），它插在相邻的 FET 之间提供**电绝缘**。玻璃的作用是保证两个晶体管之间没有电流通过，保持它们在电气上是隔离的。值得再次提出的另一点是 n 型和 p 型区之间的 pn 结有阻断电流的能力。因此可以假设 n 区和 p 区间是电绝缘的。[7]

[7] 阻断电流的能力要求 n 边的电压高于 p 边的电压。

本例的顶视图见图 3.24。图中仅清楚显示的工艺层是 n 阱，n+（nFET 漏/源），p+（pFET 漏/源和栅(多晶硅)。p 衬底和氧化层隐含存在。注意，FOX 存在于每个晶体管周围，除了晶体管位置，其他所有的地方都隐含有场区存在。FOX 区很少明确画出，所以一定要记住圆片中的每个器件自然都与每个其他器件隔离。

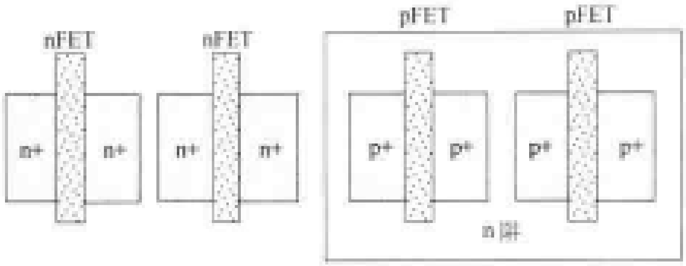


图 3.24 FET 图形顶视图

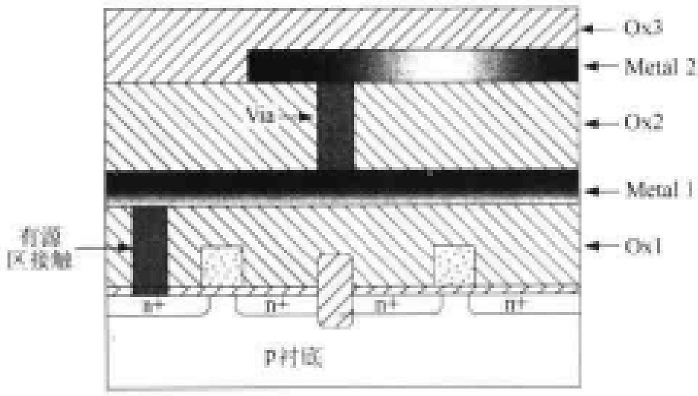


图 3.25 金属互连层

一旦定义了基本的晶体管工艺层，就可加上用玻璃绝缘体分隔开的导电金属层以便连线。现代工艺往往可有 5 个或更多的金属互连线层以缓解复杂电路中大量布线的问题。图 3.25 的例子用两个金属层来说明要点。在 FET 形成后，将一层氧化层(Ox1)淀积到圆片表面并使之平面化。然后在氧化层上刻蚀一个“孔”(称为接触孔)以便能与漏/源区进行电连接，这在图中显示为有源区触点，孔里填充像钨这样的导电金属。然后将金属层 1 淀积于上，接着是另一绝缘的氧化层(Ox2)。注意，利用在 Ox1 上刻蚀的孔，也可以将第一层金属(Metal 1)与栅层相连。然后再将第二层金属(Metal 2)淀积到 Ox2 的上面。Metal 1 和 Metal 2 之间的电连接是通过通孔(Via)完成的。这是一个在 Ox2 上刻蚀的孔，其中填充以导电的金属“插塞”。

既然看到在 CMOS 工艺中金属互连层是如何加上去的，那么以下的观察是非常重要的：

- 金属层之间以及金属层与晶体管之间用玻璃实现电绝缘；
- 相邻导电层之间的电接触要求在它们之间的氧化层上形成**接触孔**和**通孔**。

这意味着不需要在导电层之间建立电通路就可以穿越它们。这些规则的例子见图 3.26 的版图(顶视图)。Metal 2 的线可穿越其他任意一层；与 Metal 1 的连接需要一个通孔。用栅接触孔可将 Metal 1 与栅连接，而 Metal 1 与 FET 的漏/源区的电连接则是由一个有源区接触实现的。

CMOS 电路设计首先在硅上形成 nFET 和 pFET，然后通过在导电层上形成的互连线将它们连接在一起。有意思的是，数字 CMOS 逻辑电路仅仅是由晶体管和导线构成的。不管系统多么复杂，也不需要任何其他器件。一旦学会如何设计基本的 FET 及如何加上互连线，就掌握了 CMOS VLSI 的基础。

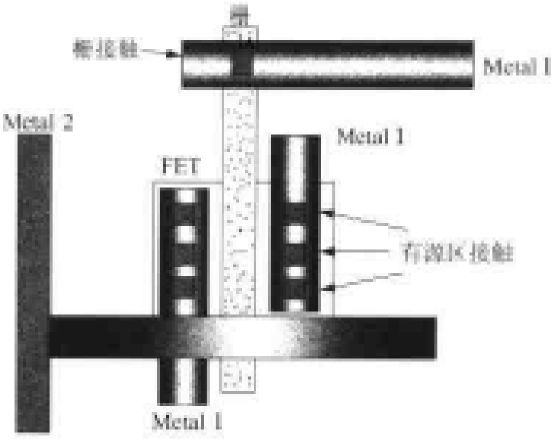


图 3.26　互连线的版图例子

3.4　FET 阵列设计

CMOS 逻辑门是由输入变量控制的开关电路。这些开关阵列采用 FET 用导线连在一起，形成串、并联组，其连接方式能够得到所需要的功能。在第 2 章中我们学习了在电路层次上如何用 FET 建立逻辑门，它们最终必须转化为硅片图形才能完成最后的设计。在 VLSI 中，图形本身变成了电路。用形成图案的多边形图形，跟踪信号和电压初看起来可能有点奇怪，但将很快学会如何去读懂它们代表的逻辑流及操作。

让我们从一个最简单的 n 型管子堆叠的例子开始。它是由两个 nFET 串联组成的。图 3.27(a)是该例子的线路图。信号 A 和 B 分别加在两个管子的栅极上。注意，在硅片上建立图形实际只需要三个 n+ 区：左边一个，中间一个，右边一个。这一简单的观察可以画出如图 3.27(b)的两个晶体管组合的硅片图形。这是关于高密度集成电路的第一课：

器件可以共用图形区域，以节省版图面积或降低复杂性。

就目前的例子，并不需要先建立一个个的器件，然后再把它们连在一起。如果将 n+ 区合在一起，可以得到一个更有效的设计。图 3.27(c)侧视图表明，这个技术确实建立了一个由两个晶体管组成的信号通路，只有在两个晶体管都导电时才能从左至右导电。

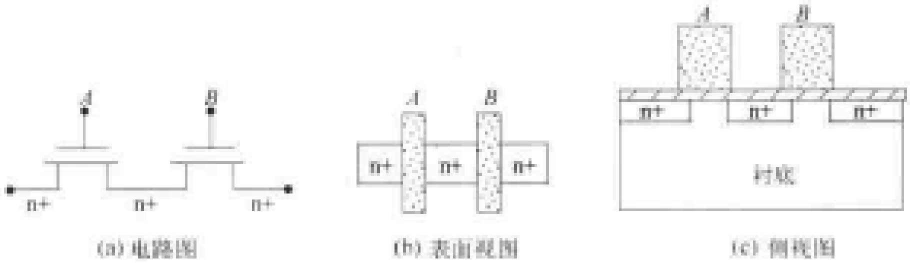
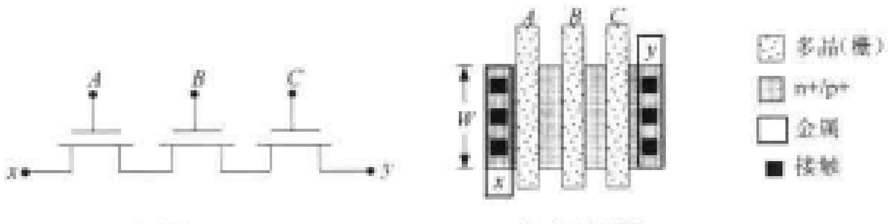


图 3.27　两个串联 nFET 的硅片图形

这一技术适用于任何 FET 串联组合。图 3.28 是一个 3 个 FET 的链。在表面视图上并没有标出每个区域，比较方便的是提供一个图例，用不同的填充样式表示不同的材料。如用颜色表示各工艺层更为简便，它是计算机辅助设计中更倾向使用的技术。图中左右两边加上金属线以

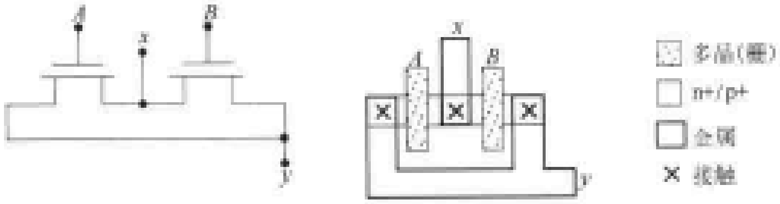
及有源区接触(连接金属到 n+区)。如图,这些金属线和接触确定了到节点 x 和 y 的电气连接,且将这个晶体管组与电路的其他部分相连接。图形还显示了三个晶体管的沟道宽度 W,因而比起图 3.27 的简单表面图形来提供了更多的信息。在最初设计阶段,沟道宽度并不总是明确标出的。在设计的这个阶段,通常对信号流路径和电路拓扑连接比对晶体管细节更感兴趣。换句话说,我们希望在处理选择晶体管实际尺寸这样的细节之前先进行设计和验证逻辑。



(a) 电路图 (b) 表面视图

图 3.28 三个串联的 nFET

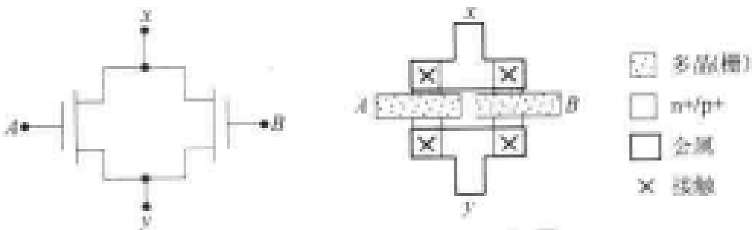
并联的 FET 可用同样的方式形成图形。图 3.29 中,用金属图形将两个 nFET 并联起来。这一并联连接可从两个晶体管的漏区/源区都连接在节点 x 和 y 之间看出,这意味着它们并联在一起。图 3.29(a)为线路图,3.29(b)为晶体管图形和布线示意图。这个形成并联 FET 表面图形的方法保留了上面在串联 FET 图形中晶体管的方位。这可能是我们所期望的:一种统一的版图原则可以使硅表面上有较高的集成密度。



(a) 电路图 (b) 表面视图

图 3.29 并联 FET 图形

图 3.30 是并联 FET 的另一种版图方案。它将晶体管漏-源位置改为垂直方向。在这种方法中,两个 FET 由分开的 n+区构成。并联连接是通过金属互连线连接节点 x 和 y 完成的。虽然这两种方法使两个 FET 位置一致(水平或垂直),但这并不是必需的。只有布线和最终的电连接才是重要的。互相分开的晶体管通常比共享漏/源区的晶体管占用更多的面积,所以这种类型仅限于特殊情况。



(a) 电路图 (b) 表面视图

图 3.30 并联 FET 的另一种版图方案

3.4.1 基本门设计

既然已经了解了有关 CMOS 版图设计的基本概念，现在让我们来考察一下硅片上 CMOS 逻辑门所使用的表面图形。在本节中先将问题简化，把导电层上的图形线看成引导电流和建立电压的路径。在这个层次上，线的宽度并不重要，只需要用电路中的连接关系来跟踪逻辑。这个方法在 CMOS 版图设计的起始阶段非常有用，因为它可以使设计者通过改变器件的位置和方向来看它们的排布是否合理。

先让我们看一个非门。图 3.31(a)表明如何用晶体管 Mn 和 Mp 作为互补对连接成一个电路的。图 3.31(b)是在硅片上的实现。由于已完成版图，可以看到它与电路的一一对应情况，其中有几个重要的特点：

- 电源(V_{DD})和地(Gnd)用金属线层布线
- n+ 和 p+ 区用同样的填充图案表示，不同的是 pFET 嵌在 n 阱的边界内
- 由于金属和 n+ 或 p+ 区处于不同的结构层，所以从金属层至 n+ 或 p+ 区需有接触孔

跟踪版图上逻辑操作的能力是一个非常有用，需要发展的技巧。本例中，输入 x 控制多晶栅。当 $x=0$ 时，Mp 像一个接通的开关而 Mn 则断开，输出为 V_{DD}，即 $\bar{x}=1$。反之，输入 $x=1$ 使 Mn 导通而 Mp 断开。这使 Gnd 连接到输出，它相当于 $\bar{x}=0$。

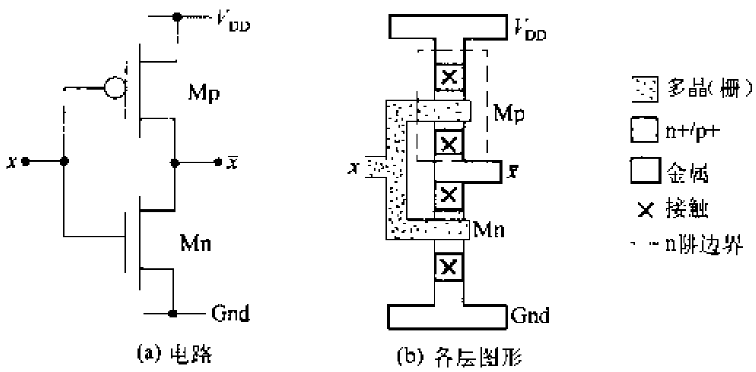


图 3.31 非门(NOT)电路转换成硅片实现

图 3.32 是另一种方案的版图。这时，把非门画成像一个 2:1 的多路选择器。虽然操作是完全等效的，但从与多路选择器电路一一对应得到的版图中，可以看到 FET 的方位与图 3.31 中的 FET 成直角。这说明，CMOS 电路可以采用不同的几何版图。如果不考虑图形的实际尺寸，版图布局的变化并不重要。物理设计的这一特点将在本书后面讨论。

物理设计的目标之一是使整个芯片的面积最小。这可以在各个层次上用各种技术来实现。图 3.33 是例子之一。图中两个非门电路共用 V_{DD}和 Gnd 连线。左边反相器输入为 a 产生$\bar{a}$，而右边的电路则将 b 反相为$\bar{b}$。显然，这比用两个分开电路实现的简单做法要节省面积。当然，设计必须有把两个反相器在逻辑链中紧靠在一起的需要。同样的版图也可用来作为设计图 3.34 中非反相缓冲器的基础。图 3.34(a)用了两个串联的反相器来实现这一逻辑。虽然输入 a 产生与 a 相同的布尔逻辑值，但缓冲器提供了电信号的整形并为大的扇出负载提供额外的“驱动强度”。图 3.34(b)中的版图示意图用左面的反相器的输出作为右面反相器的输

入。它要求两级之间有一个金属到多晶的接触孔。[⑧] 该图还利用了金属可以跨越输入多晶栅而不会与之在电气上连接这一事实。

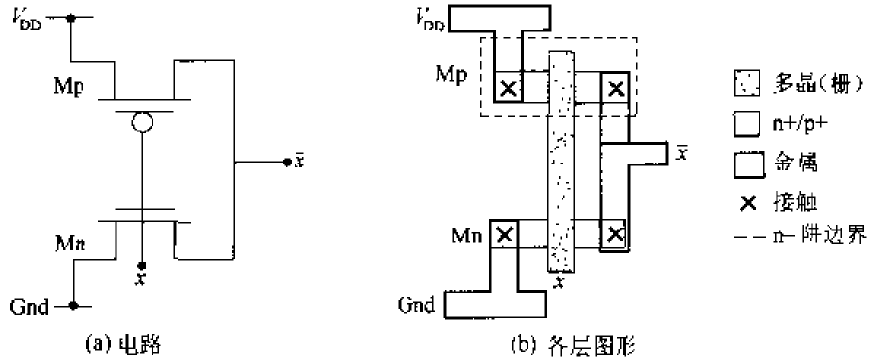


图 3.32 非门(NOT)的另一种版图

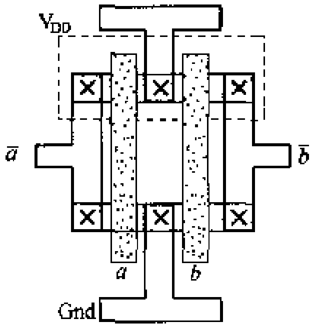


图 3.33 共享电源和接地的两个非门

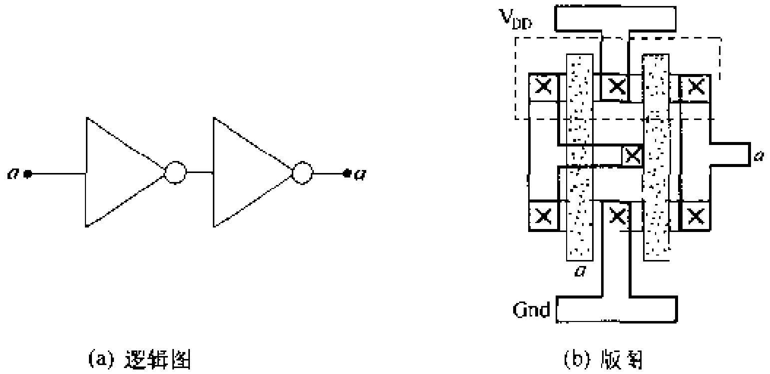


图 3.34 非反相缓冲器

传输门的问题说明版图中出现的某些互连线布线问题。图 3.35(a)中的逻辑图是一个输入为 x 输出为 y 的传输门(TG)。由于一个传输门只含两个 FET,所以在物理层次上的设计是非常简单的。问题复杂在反相器上。它接受开关信号 S 并且必须产生 $\bar{S}$ 来驱动 TG 另一边的 pFET。非门必须与电源和地连接,而 TG 的 nFET 和 pFET 的位置则可根据需要放置。图 3.35(b)是一种解决办法。它采用的反相器 FET 具有一条较长的 n+ 区,x 的 TG 金属输入线可以跨越它。互补开关信号 $\bar{S}$ 直接由反相器送入 TG pFET。

一旦建立了简单版图的基础,可以把它用在更复杂的门上。图 3.36(a)是一个二输入与非门(NAND2)电路,其画法可直接变为图 3.36(b)中的工艺层图形。两个 nFET 串联在一起,因而可按图 3.27 的方法布置。由于输入为 a 和 b 的门垂直布置,加入并联的 pFET 可以采用图 3.29 介绍的用金属导线实现并联连接的技术。这可保持栅多晶线比较简单。用同样的方法可以用来构成二输入或非门(NOR2)。图 3.37(a)所示 FET 的连接正好相反,nFET 并联而 pFET 串联。它的版图如图 3.37(b)所示,与前面 NAND2 门的布线方法采用同样的原则。

将 NAND2 和 NOR2 的结构分解为晶体管和导线,可以看到二者版图的相似性。这两个

⑧ 这称为多晶接触孔,它定义了在氧化层上的刻孔。

门 FET 的基本布置见图 3.38(a)。为了构成 NAND2 门我们采用了图 3.38(b)的金属布线图形。NOR2 门的构成是采用了图 3.38(c)的布线方式。如果稍微研究一下两个门的金属线图形,就会发现它们是一致的。要验证这一点,可以通过其中一个门的中间画一条想像的水平线,然后绕着它旋转图形(即使它转成垂直)。这表明了 AND－OR 的对偶特性如何演变成版图的对称性。

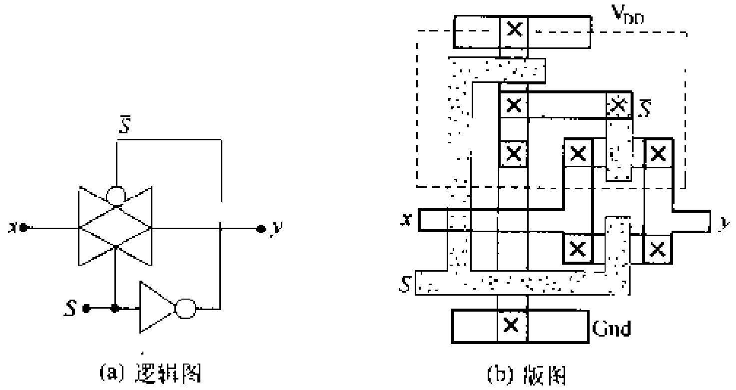


图 3.35 带有驱动器的传输门版图

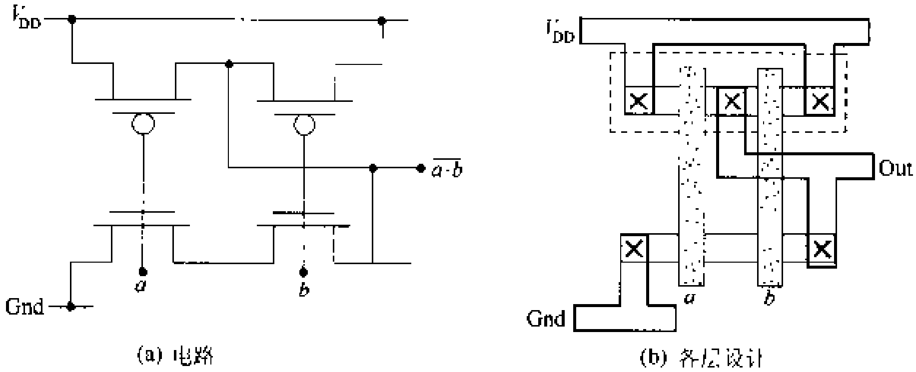


图 3.36 NAND2 版图

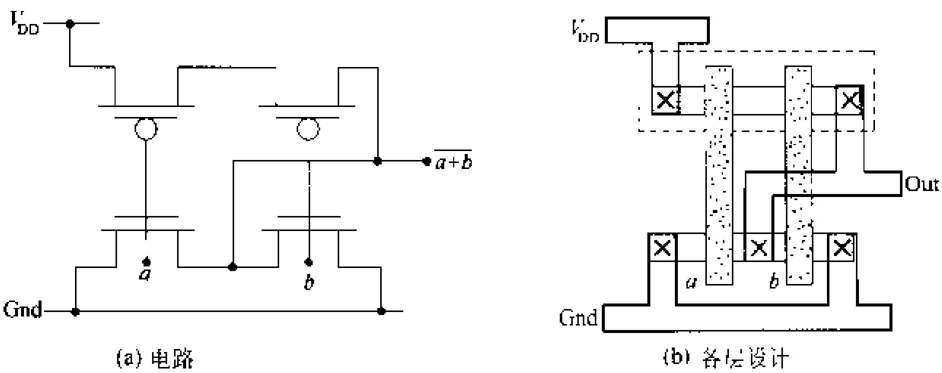


图 3.37 NOR2 门的设计

这些版图技术可以扩展到具有三个或更多输入的逻辑门。图 3.39(a)是一个三输入或非门(NOR3)。它用了三个串联的 pFET 和三个并联的 nFET。如果翻转该金属线图形,即可得

到图 3.39(b)的三输入与非门(NAND3)电路。用同样的方式也可以设计出 4 输入的门。但是,4 输入的与非和或非门的电开关时间相当慢,所以妨碍了它们的使用。

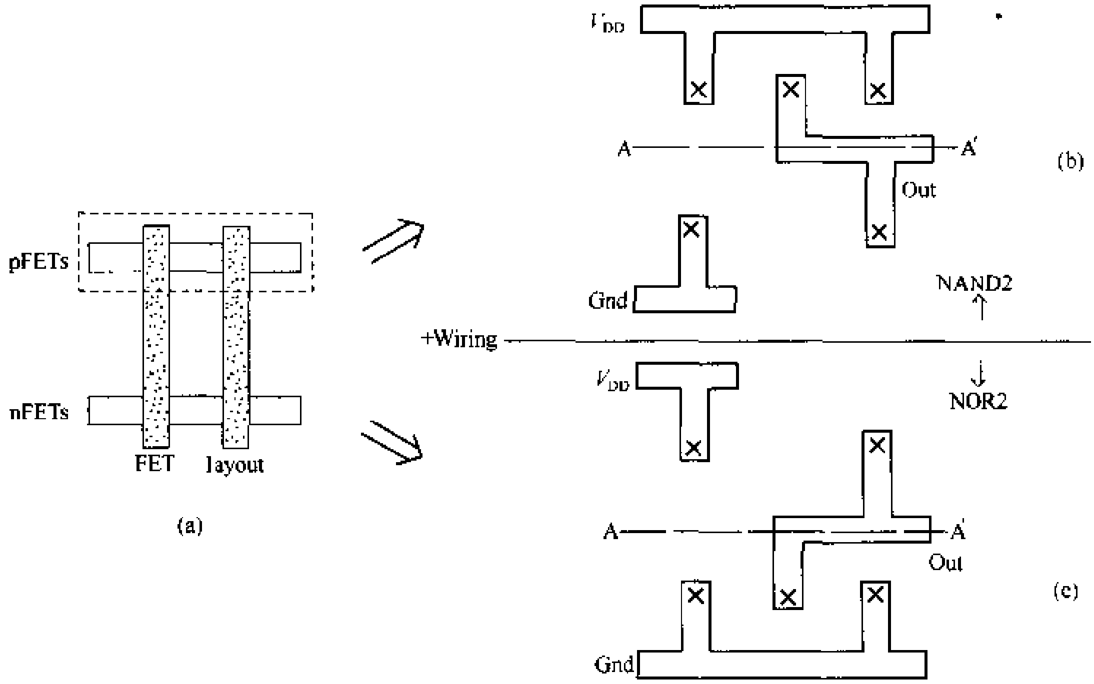


图 3.38 NAND2 – NOR2 的版图比较

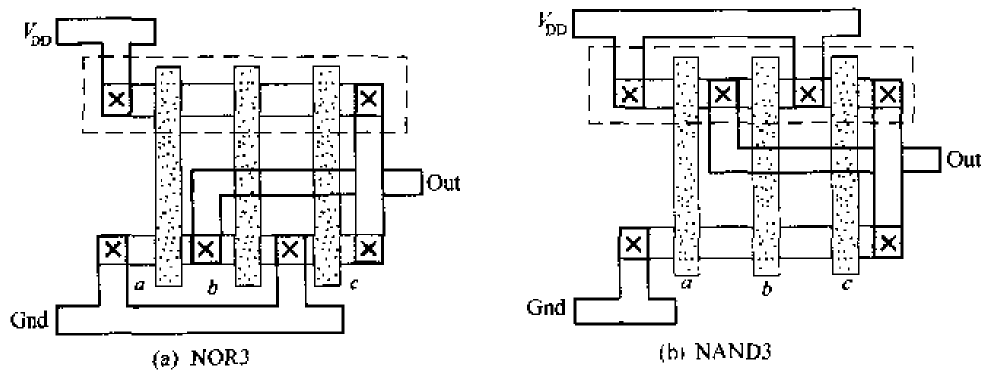


图 3.39 三输入门的版图

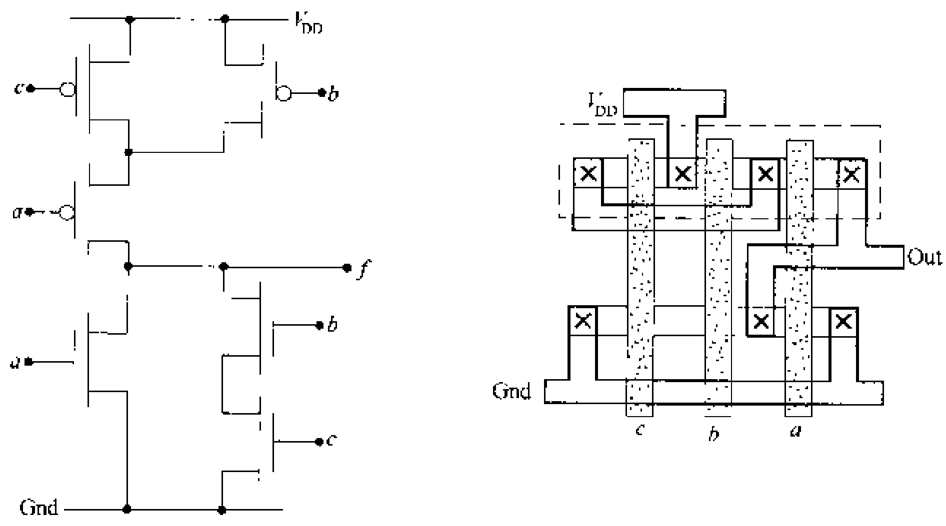
3.4.2 复合逻辑门

复合逻辑门的版图也可以用同样的方式来完成。考虑图 3.40(a)中的电路,其实现的功能

$$f = \overline{a + b \cdot c} \tag{3.68}$$

可用一般的分析方法进行验证。电路要求将一个 nFET 与一组两个串联的 nFET 并联。pFET 阵列由两个并联的一组管子组成且与另外一个器件串联。图 3.40(b)中的版图给出正确的布线图,并且每个输入都用一个多晶栅图形。注意,信号的摆放顺序是得到正确逻辑输出

的关键。



(a) 电路　　　　　(b) 版图

图 3.40 版图技术推广到复合逻辑门

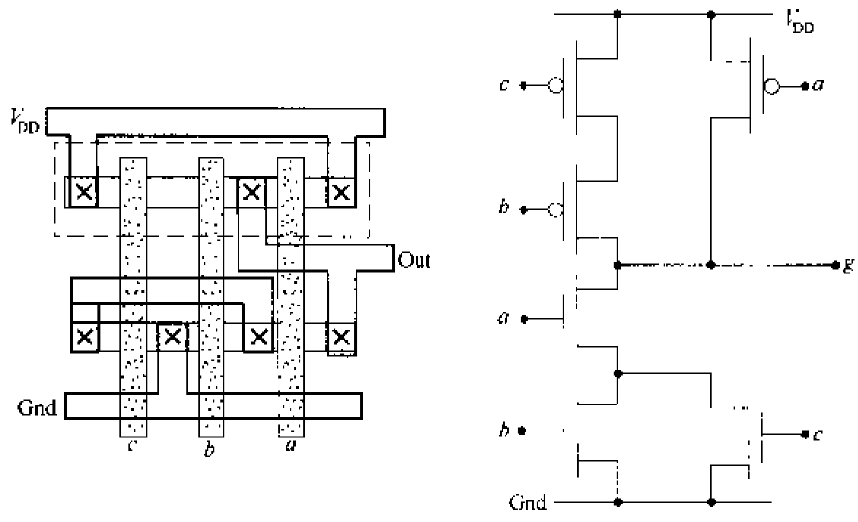
版图的一个有趣的改变表现出另一个重要问题。假设将金属布线图形围绕着想像的一条水平线翻转,所得到的版图如图 3.41(a)所示。跟踪这个电路得到如图 3.41(b)所示的电路图。可以看到新电路实现的功能为

$$g = \overline{a \cdot (b + c)} \tag{3.69}$$

它可以看成 f 的逻辑对偶。这一关系与前面讲到的或非 - 与非门(NOR - NAND)的关系是一样的,它说明了许多逻辑的对称现象可直接反映在版图上这一事实。

可惜的是,并不是所有门的版图都像上面这些例子这么简单。许多版图需要更多的思考,并且可能经历多次尝试和改动的过程来完成最后的设计。考虑 AOI 的一般表达式

$$F = \overline{x \cdot y + z \cdot w} \tag{3.70}$$



(a) 版图　　　　　(b) 电路

图 3.41 构成对偶电路

该功能可以用图3.42(a)中的电路来实现。若想保留每个输入都用垂直多晶线的版图策略，先画出四条多晶栅线以及 V_{DD}和 Gnd 线。为使面积最小，需要共用 n+ 和 p+ 区。画 nFET 图形较为简单，它由两组管子组成，每组各含两个 nFET，见图 3.42(b)版图。这里用粗线表示金属线，因为在最初的设计中只有布线是重要的。一旦放置好了 nFET，pFET 就必须按照电路的要求正确地连接。对于这个门，版图上所示的 pFET 连接是一个正确的结果。注意，在电路图中，z 和 w pFET 组与电源连接，而在版图中画的是 x 和 y 与 V_{DD}相连。二者都为 V_{DD}和输出 F 之间提供同样的开关特性，所以该版图是允许这样画的。

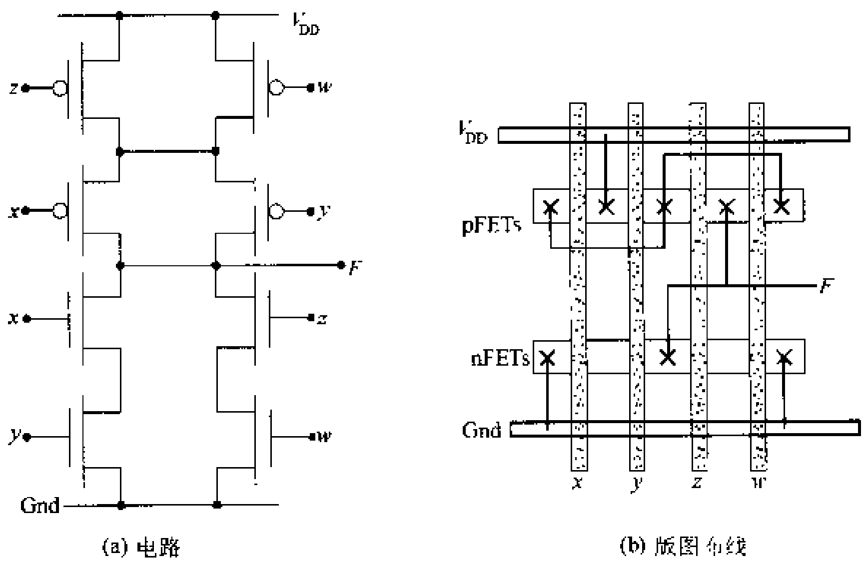


图 3.42 一个一般的 4 输入 AOI 门

3.4.3 一般性讨论

前面这些例子说明了建立门级版图的一些基本技术。在考察过的基本门中，几个晶体管之间共享 n+ 和 p+ 区是可能的，它减少了面积和布线的复杂性。但这并不总是可行的，特别是对一些复杂的情况。许多年来发展了各种布置 FET 和布线的方法，值得在此进行讨论。

考虑在 CMOS 电路中布置晶体管的一般性问题。经验表明，规则的图形和阵列可以得到最高的密度，随意放置的多边形应尽可能避免。一般来说，每个逻辑门都要求有与电源(V_{DD})和地(V_{SS})的连接，在我们的例子中，它们布置成水平方向的金属线，一般也都是这样布置的。这样，形成了如图 3.43 中的基本框架。所有的 FET 都放置在两个供电轨道(V_{DD}和 Gnd)之间。图中的晶体管有的是单独的器件，有的则组合在一起共用栅多晶线或共用漏/源区。后一种情形的布置最节省面积，但并不总能与晶体管这样连接。图中还显示了栅线可以垂直或平行于电源线。虽然图中没有明确画出，但 pFET 将嵌在 V_{DD}附近的 n 阱内，而 nFET 则靠近地线。

有一种画版图的办法以简单**棍棒图**的概念为基础，即用不同的颜色代表不同的工艺层，布线为有颜色的线并且服从构成芯片的规则。图 3.44 是棍棒图的一个简单例子。为了节省印刷费用以使本书的价格尽可能低廉，图是黑白的，用不同特征的线条(如宽度不同或用虚线)来代表各工艺层。图例说明各条线所对应的工艺层。图中列出了每个工艺层最常用的颜色。它们是：

- 多晶硅(栅):红色
- 掺杂 n+/p+(有源区):绿色
- n 阱:黄色(可以用其他颜色)
- 金属 1:蓝色
- 金属 2:灰色(可以用其他颜色)
- 触点:黑色的叉号(╳)

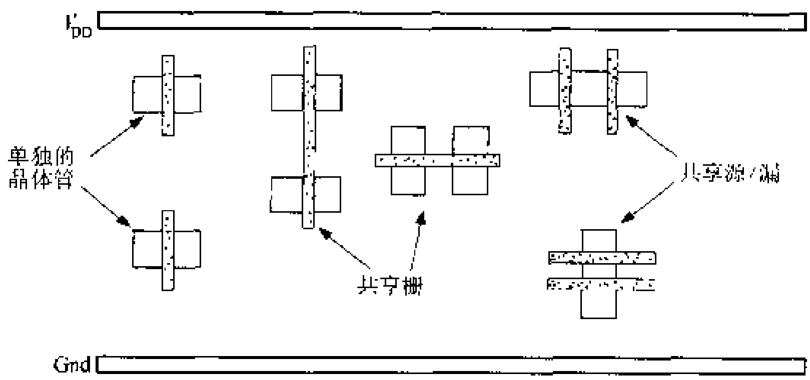


图 3.43　逻辑门版图的一般几何图布置

准备一套颜色铅笔,版图设计者就能很容易画出和验证版图以最终转移到硅片上。下面是与彩色棍棒图有关的一些简单规则:

- 红线与绿线交叉产生一个晶体管
- 黄色边框区内红色在绿色之上为 pFET 管;不在黄色框内为 nFET 管
- 红色可以越过蓝色或灰色
- 蓝色可以越过红色、绿色或灰色
- 灰色可以越过红色、绿色或蓝色
- 从蓝色到绿色必须放置晶体管的接触孔
- 蓝色连接绿色必须通过通孔
- 蓝色连接红色必须使用多晶接触孔

这套简单规则是棍棒版图的基础。本书附带的 CD 盘可用彩色显示屏对棍棒图进行更详细的讨论。棍棒图常用于快速完成版图或用于研究较大的复杂布线问题。

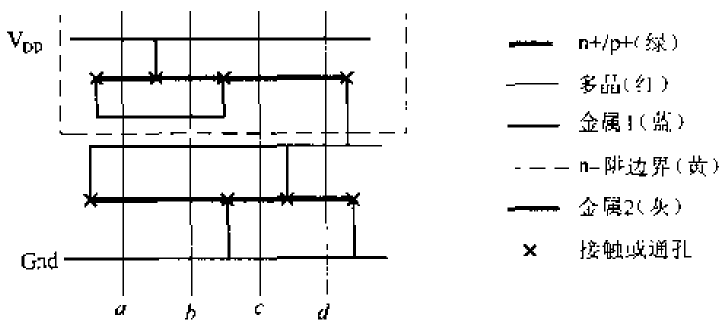


图 3.44　基本的棍棒版图示意

一个更为结构化的技术是将图论应用于晶体管布置和逻辑门版图的问题。图 3.45 定义代表 FET 管的图示单元的基本组成部分。用这种方法，晶体管漏和源的节点 x 和 y 相当于连接的节点称为**顶点**。晶体管本身用**边**来表示，它对应于信号流路径。任何一个 CMOS 电路都可转化为一个由边和顶点组成的等效图。

欧拉(Euler)图在晶体管共用漏/源区时有助于电路的布置和布线。为了建立一个欧拉图，先从 CMOS 电路图开始选择一个起始顶点(节点)。可能的话，从该点开始走过整个图形，使每边只能通过一次，若能这样，nFET/pFET 就可共用 n+/p+ 区。这样完成的图可以直接用来建立版图策略。

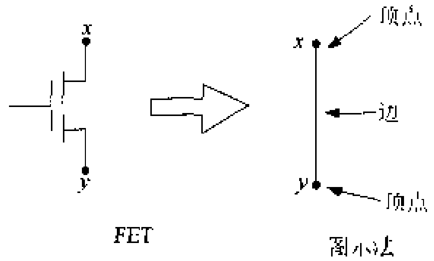


图 3.45 图论中 FET 的表示法

图 3.46 就是这个过程的一个例子。如图 3.46(a)所示可以跟踪图中的电路。路径起始于所示的一个顶点，并顺箭头到达终点，每条边只通过一次，这就定义了**欧拉路径**。由于存在这样的路径，可以用它来构成图 3.34(b)中的欧拉图。该图由相交的 pFET 图和 nFET 图构成。pFET 图将 V_{DD}与节点 α 相连，再连接到输出节点 OUT，用输入变量来标记每一条边。nFET 图与每一条 pFET 边相交一次，所得到的路径即为 nFET 链。注意，nFET 和 pFET 图都是封闭的，由此可以断定每一种极性的管子都可以共用一个 n+/p+ 区。为了将欧拉图变为版图，我们可以沿晶体管路径按图中的顺序来连接 FET。本例中的版图如图 3.47 所示。FET 由图左边定义的简化符号代表。一组 pFET 串在一起，按欧拉图中 pFET 部分表示的布线法可转变为版图。同样，我们用以共用 n+ 区来形成 nFET，并按欧拉图的 nFET 部分布线。然后将其转化为最终的版图。

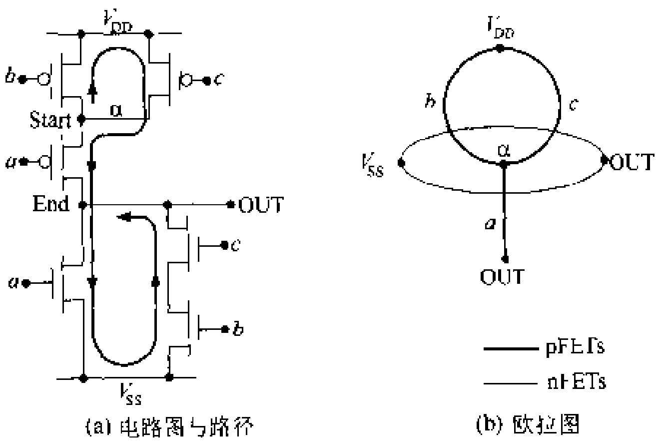


图 3.46 欧拉图的构成

如果找不到一条欧拉路径，说明不能用 FET 链构成电路。此时需要两个或更多的晶体管

组，其版图也将复杂得多。一些自动设计工具已被开发，在某些方面帮助逻辑门版图的设计，但在关键的应用中仍然需要考虑有经验的版图设计者。许多版图专家都具有图画或艺术方面的知识背景，并且能够创造出其他人很难想到的巧妙紧凑的设计。

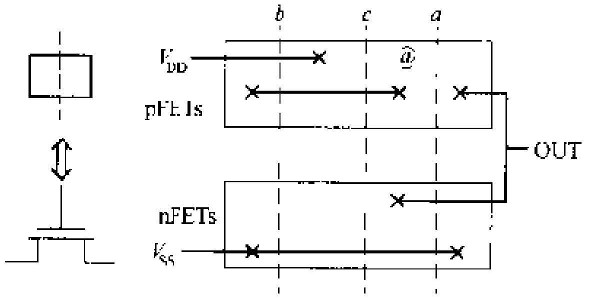


图 3.47 用欧拉图画版图

3.4.4 小结

通过本章了解了将 FET 逻辑电路转移到硅片上的基本知识。本章介绍的有关版图的各种考虑足以用一组标准的 MOSFET 作为积木块，构成复杂的逻辑电路。在许多设计中，只要简单地将版图规定的合适尺寸的晶体管放置好，且将它们连接在一起就可以了。如果一切无误，就可得到一个逻辑功能正确的电路。但是电路的开关速度也许不会如所期望的那么快。

高速 VLSI 设计的关键是建立一个能够尽可能快地完成所要求操作的开关电路。这就意味着，必须开始考虑由晶体管开关时间和寄生电阻与电容元件造成信号延迟的问题。这使我们进入到 VLSI 设计者的独特世界。不能仅仅满足于只得到一个能工作的电路，它还必须是快速的。

CMOS 生产工艺的细节和它对电气性能的影响是在本章讨论的内容和高性能系统设计之间缺少的一环。下面几章所引入的概念将加强和扩展本章的内容。于是，CMOS 逻辑电路、版图、晶体管和系统设计之间的关系就很自然了。

3.5 参考资料

[1] H. B. Bakoglu, **Circuits, Interconnections, and Packaging for VLSI**, Addison-Wesley, Reading, MA, 1990.

[2] Dan Clein, **CMOS IC Layout**, Newnes, Woburn, MA, 2000.

[3] Richard S. Muller and Theodore I. Kamins, **Device Electronics for Integrated Circuits**, 2nd. ed., John Wiley & Sons, New York, 1986.

[4] Robert F. Pierret, **Semiconductor Device Fundamentals**, Addison-Wesley, Reading, MA, 1996.

[5] Bryan Preas and Michael Lorenzetti (eds.), **Physical Design Automation of VLSI Systems**, Benjamin/Cummings Publishing Company, Menlo Park, CA, 1988.

[6] M. Sarrafzadeh and C. K. Wong, **An Introduction to VLSI Physical Design**, McGraw-Hill, New York, 1996.

[7] Naveed Sherwani, **Algorithms for VLSI Physical Design Automation**, Kluwer Academic Publishers, Norwell, MA, 1993.

[8] Jasprit Singh, **Semiconductor Devices**, John Wiley & Sons, New York, 2001.

[9] Ben G. Streetman and Sanhay Banerjee, **Solid State Electronic Devices**, 5th ed., Prentice Hall, Upper Saddle River, NJ, 1998.
[10] John P. Uyemura, **Physical Design of CMOS Integrated Circuits Using L-Edit™**, PWS Publishers, Boston, 1995.
[11] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.

3.6 习题

[**3.1**] 考虑图 P3.1 中的互连线图形。线的宽度为 1 个单位，薄层电阻 $R_s = 25\ \Omega$。如果每个拐角方块的电阻是直路方块电阻的 0.625，求由 A 到 B 的总电阻。

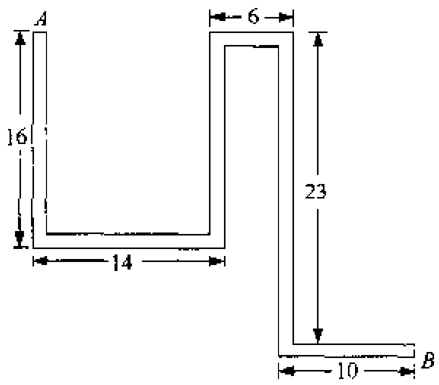


图 P3.1

[**3.2**] 互连线可用两个工艺层中的任意一个。如果选用栅多晶层，其薄层电阻为 25 Ω，这时互连线的宽度为 0.5 μm 而长度为 27.5 μm。也可以用金属层，其薄层电阻为 0.08 Ω，金属线宽 0.8 μm，设要求的布线长度为 32.4μm。

计算上述两种情况下的线电阻 R_{line}，并且选用电阻较小的一个。如果改用电阻较大的互连线，其电阻增加的百分比为多少？

[**3.3**] 有个互连线由电阻率为 $\rho = 4\ \mu\Omega\cdot\text{cm}$ 的材料制成。线厚度为 1200 Å(1 Å = 10^{-8} cm)，宽为0.6 μm。

(a) 计算线的薄层电阻 R_s。

(b) 求线长为 125 μm 时的线电阻。

[**3.4**] 考虑互连线时间常数 τ 的公式(3.14)。通过将欧姆和法拉改用基本的 MKS 单位(米－千克－秒)来表示并化简，证明 τ 的单位为秒。

[**3.5**] 有个互连线布在一个厚度为 10 000 Å的绝缘氧化层上。线宽0.5 μm，线长40 μm。已知薄层电阻为 25 Ω。

(a) 求线电阻 R_{line}。

(b) 求线电容 C_{line}。采用 $\epsilon_{ox} = 3.453 \times 10^{-13}$ F/cm，且将计算结果用飞法拉(fF)表示(1fF = 10^{-15}F)。

(c) 求该线的时间常数 τ，单位用皮秒(ps)，1ps = 10^{-12} s

[**3.6**] 一掺有砷的硅样品，$N_d = 4 \times 10^{17}\ \text{cm}^{-3}$。

(a) 求多子载体密度。

(b) 求少子载体密度。

(c) 计算电子和空穴迁移率，然后求出样品的电导率。

[**3.7**] 一硅区同时掺有磷和硼。磷掺杂浓度为 $N_d = 2 \times 10^{16}\ \text{cm}^{-3}$，而硼掺杂浓度为 $N_a = 6 \times 10^{18}\ \text{cm}^{-3}$。确定该硅区的极性(n 或 p)，并求载流子密度。

[**3.8**] 一掺有硼原子的硅样品，其受主密度为 $N_a = 4 \times 10^{14}\ \text{cm}^{-3}$。

(a) 求多子和少子密度。

(b) 求样品的电阻率 ρ。

(c) 设该区的大小为 2 μm×0.5 μm×100 μm。求该区边到边样品块的最大电阻。

[3.9] 现有一个掺杂半导体,其

$$\sigma = q(\mu_n n + \mu_p p) \tag{3.71}$$

且 $np = n_i^2$,如果希望电导率最小。

(a) 利用质量作用定律只用含 p 的项写出上式。

(b) 计算导数($d\sigma/dp$)并令其为0,求出使 σ 值最小的空穴浓度。

(c) 注意 $\mu_n > \mu_p$,若要得到最高电阻率需要什么极性(n 型或 p 型)? 然后用公式求出得到最高电阻率的掺杂类型和密度。

[3.10] 一个 n 沟道 MOSFET 的迁移率为 $\mu_n = 560 \text{cm}^2/(\text{V·s})$,栅氧厚度为 $t_{ox} = 9$ nm。已知栅电压 $V_G = 2.5$ V,阈值电压为 0.65 V。

(a) 计算 C_{ox}值,单位为 F/cm²。

(b) 求工艺互导 k'_n。

(c) 求 FET 沟道长度为 0.25 μm,宽度为 2 μm 时的器件互导 β_n。

[3.11] 利用求 R_n 的公式(3.57)求出电子迁移率 μ_n 的单位。假设已知 $\mu_n = 500 \text{ cm}^2/(\text{V·s})$,$(V_G - V_{Tn}) = (3.3 - 0.7)$V。

(a) 求当 $W = 10$ μm,$L = 0.5$ μm,$t_{ox} = 10$ nm 时 nFET 的电阻。

(b) 求当沟道宽度增加为 $W = 22$ μm 而沟道长度不变时 R_n 的值。

[3.12] 一个 pFET 管的 $\mu_p = 220 \text{ cm}^2/(\text{V·s})$,$(V_G - V_{Tp}) = (3.3 - 0.8)$ V,$W = 14$ μm,$L = 0.5$ μm,$t_{ox} = 11.5$ nm。求这一器件 pFET 的电阻 R_p。

[3.13] 工艺的氧化层厚度 $t_{ox} = 9.5$ nm。已知粒子迁移率为 $\mu_n = 540 \text{ cm}^2/(\text{V·s})$及 $\mu_p = 220 \text{ cm}^2/(\text{V·s})$。制造一个 nFET 和一个 pFET,均有 $W = 12$ μm,$L = 0.35$ μm,栅电压 $V_G = 3.3$ V,而阈值电压 $V_{Tn} = 0.65$ V 和 $V_{Tp} = -0.74$ V。

(a)求两个晶体管的 R_n 和 R_p。

(b)假设保持 nFET 尺寸不变而增加 pFET 的宽度,使 $R_p = 0.8\, R_n$,求所要求的 pFET 的宽度。

[3.14] 设计一个具有下列功能的 CMOS 逻辑门。

$$\text{Out} = \overline{x \cdot (y \cdot z + z \cdot w)} \tag{3.72}$$

然后画出该电路基本的版图。

[3.15] 设计一个实现下列功能的 CMOS 门的电路和版图。

$$F = \overline{a \cdot b \cdot c + a \cdot d} \tag{3.73}$$

要求使用最少数目的晶体管和画出一个紧凑的版图。

[3.16] 根据 OAI 逻辑功能

$$g = \overline{(a + b) \cdot (c + d) \cdot e} \tag{3.74}$$

设计一个 CMOS 逻辑门并为该电路画一个基本的版图。

[3.17] 将公式(3.74)中的函数 g 展开成 AOI 形式。然后设计 CMOS 逻辑电路和版图。

[3.18] 考察图3.44中的棍棒图。这是否是一个具有功能的逻辑门。如果是,确定其提

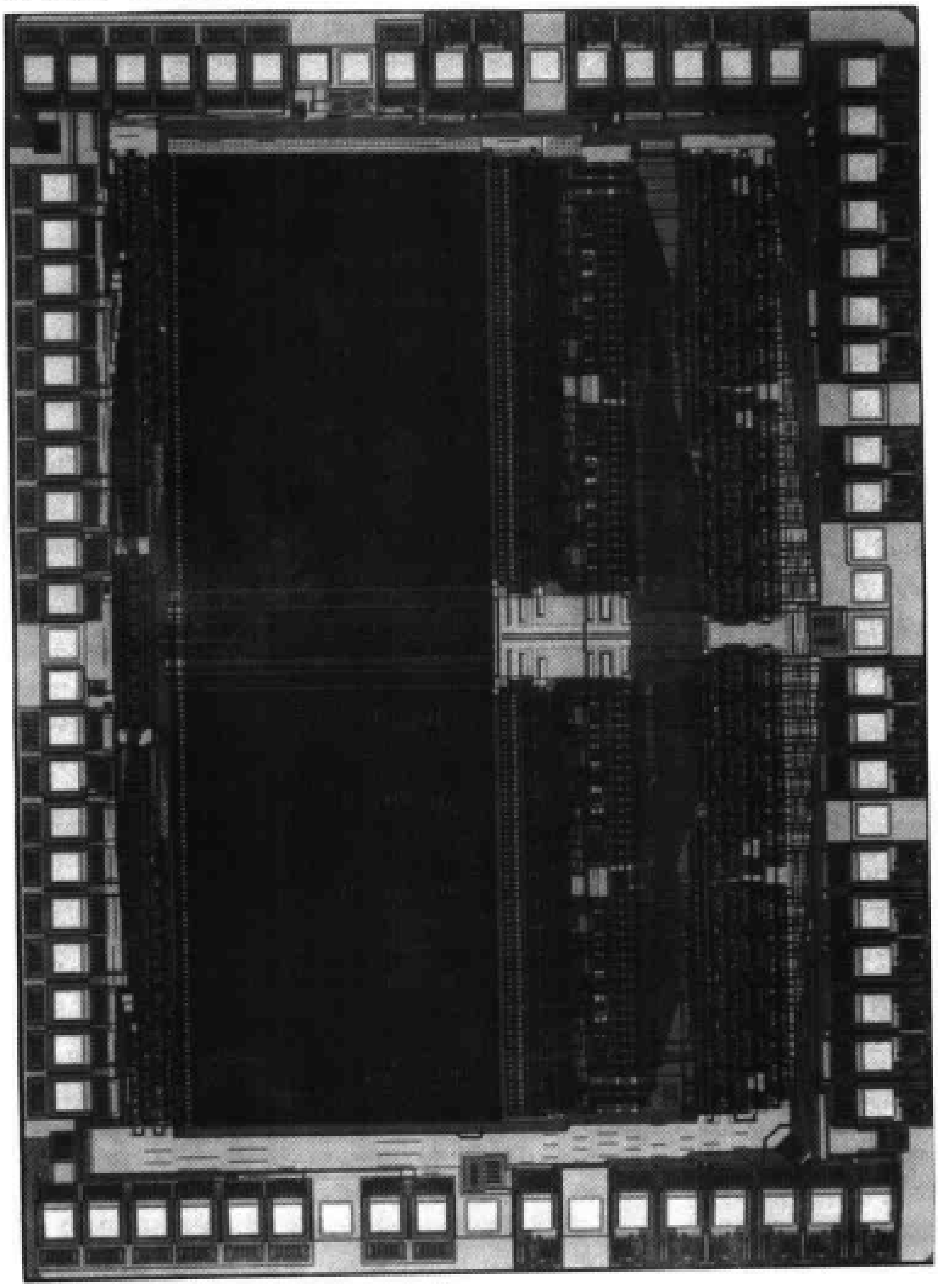
供的逻辑操作。

[3.19] 考虑下列逻辑功能

$$g = \overline{a \cdot b \cdot c + d} \tag{3.75}$$

(a)设计一个提供上述功能的 CMOS 逻辑门。

(b)能否找到该电路的欧拉图？如果能，画出该图且用它完成一个棍棒型的版图。如果不能，为该门找到一种版图策略。



第4章 CMOS集成电路的制造

集成电路是由若干层具有一定图案形状的材料层构成的,这些材料层用来形成电路中的晶体管以及电气上的互连。就现代工艺而言,最小的特征尺寸大约已经小于0.12 μm,这使集成电路可以达到极高的封装密度,单个芯片含1亿个以上的场效应管(FET)正变得非常普遍;以巨资开发制造如此复杂硅芯片所需要的技术已经进行好几十年,事实上硅被认为是在地球上研究得最多的元素!

既然了解了CMOS集成电路的物理结构,现在就可以进一步研究它们在生产过程中是如何制造的。研究将集中在硅芯片制造中对VLSI设计十分重要的一些方面。

4.1 硅工艺概述

硅集成电路是在称为圆片(wafer)的较大圆形硅薄片上制造的。圆片的直径一般为100~300 mm,厚约0.4~0.7 mm。一个规模较大的硅电路每边大约1 cm,所以在一个圆片上可以制作许许多多这样的电路。一个这样的电路所在的位置称为芯片部位(site),一个圆片上芯片部位的数目取决于每个部位的尺寸以及它与整个圆片表面积的相对关系。图4.1画出一个圆片及它上面的各个芯片部位。圆片的平口(flat)作为一个假想栅格的参考面,各个芯片部位就是放置在这个假想栅格上的。有些圆片还会有附加的平口,一看就可知道有关晶体取向信息。

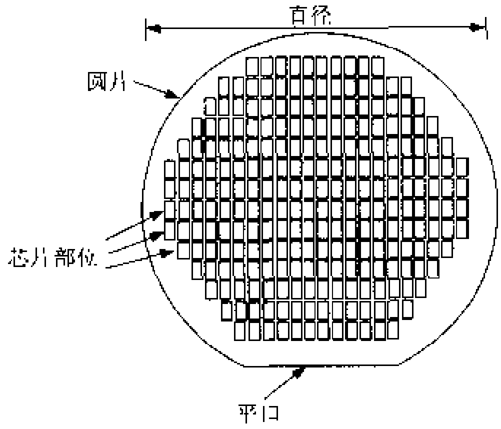


图4.1 硅圆片及其芯片部位

在集成电路制造过程中,圆片从抛光的裸表面开始需要经过几千个步骤。在这一系列步骤中,最重要的几个步骤是用来形成CMOS结构所需要的材料层及其图案。其余大多数步骤是清洗(cleaning and rinsing)圆片。一个芯片制造厂的生产能力通常是用每星期**投入圆片**的数目来衡量的,也就是每星期有多少新的圆片投入生产流程中去。圆片是分组进行加工的,一批圆片经过整个工艺线需要几周的时间。

可惜的是,并不是圆片上的每个芯片部位最后都能成为一个功能正确的电路。这是由生产线中可能发生的许多因素造成的,它们在本质上源于工艺过程的复杂性。为了说明这个问题,我们引入生产**成品率** Y 的概念,即

$$Y = \frac{N_G}{N_T} \times 100\% \tag{4.1}$$

式中 N_G 是好的(功能正确的)芯片部位的数目,而 N_T 是芯片位的总数目。成品率 $Y = 85\%$ 是指 85%的芯片可以按要求正确工作而销售给用户。自然希望有高的成品率值以帮助确保公司经济的稳定。然而提高成品率是一个非常复杂的问题,它需要对工艺线进行相当长时间的研究和实验。

成品率分析的基础是预测一个具体工艺的成品比率,它需要全面理解硅工艺过程的所有特点。这个领域的工作人员面临的问题,是对一个已定设计如何提高它的 Y 值。影响提高成品率至关重要的一个因素是芯片的面积 A_{die}。在直径为 d 的一个圆片上总的芯片部位的数目 N_T 可估计为

$$N_T = \pi \frac{(d - d_e)^2}{4A_{die}} \tag{4.2}$$

式中 d_e 为由于把矩形的芯片部位放在圆形的圆片上所浪费的周边距离。经验分析表明面积大的芯片成品率会较小,这可以用以下简单的式子来表示:

$$Y = e^{-\sqrt{DA}} \tag{4.3}$$

式中 A 是芯片的面积。参数 D 是**缺陷密度**,单位为 cm^{-2},它是圆片上每平方厘米的平均缺陷个数。D 表示对硅圆片所能期望的完好性的限度;这是由于每个硅晶圆片都具有不可避免的随机的不完好性。对于现代工艺,$D = 1cm^{-2}$是缺陷密度的合理值。

某些物理缺陷往往会以成团的形式出现在圆片上。当这样的缺陷为主时,成品率的估计由下式确定:

$$Y = \left(1 - \frac{A_{die}D}{c}\right)^c \tag{4.4}$$

式中 c 为考虑成团缺陷时的经验参数。这一考虑已写成如下二项式的形式:

$$Y = \frac{1}{\left(1 + \frac{A_{die}D}{c}\right)^2} \tag{4.5}$$

另一方面,如果圆片的一个较大面积 A_{fail}上只有几个芯片缺陷,那么成品率通常由以下表达式来近似:

$$Y \approx (1 - g)e^{-A_{die}D} \tag{4.6}$$

式中

$$g = \frac{A_{fail}}{A_{wafer}} \tag{4.7}$$

为有缺陷芯片的面积占圆片面积 Awafer 的比例。

成品率分析是 VLSI 制造中非常特殊的一项工作。这个领域的工作人员应当有很强的物理、普通化学与物理化学、数学、统计学或工程(化学、材料或电气)方面的背景知识,并且要与生产线及圆片分析组的成员形成团队紧密配合,以达到最大的成品率值。一个问题在它被发现和定义之前是不可能解决的。"实验设计"可以准确指出问题的所在并且找到解决的办法,

因此变得非常关键。

经济学基本要点

考察与 VLSI 电路的设计、制造和市场相关的几个重要经济因素是很值得的。设 C_{chip}为制造一个芯片的成本，C_{sell}是销售价。于是每个芯片的获利为：

$$获利 = C_{sell} - C_{chip} \tag{4.8}$$

维持一条生产线必须满足：

$$获利 > 0 \tag{4.9}$$

虽然这似乎是再明显不过的事，但 VLSI 的设计者必须认识到，无论 C_{chip}还是 C_{sell}都不是很容易计算的。芯片的制造成本包括材料，所有员工(设计、制造、测试等)的工资以及其他开销(电费、水费、税款等)。提高成品率将降低每个单元的总成本，因此成品率分析的重要性十分明显。这些因素以及其他许多因素都会影响 C_{chip}。

在现代的 VLSI 中，建设一个最先进的芯片制造工厂的成本大约为10亿～30亿美元之间。这包括土地、厂房、设备及启动费，但不包括材料和日常运行费。设施费用必须按工厂的寿命期摊派到生产线上。

每个产品的销售价必须包括所有直接和间接的成本。供求规律也在其中起作用：C_{sell}必须是顾客愿意支付的价格。如果一个产品的需求量很大，那么 C_{sell}可以比成本高出很多，因此这一设计就会获取很大的收入。在这种情形下，芯片(以及通常所说的产品)可以完全按需求决定价格来销售；描述这个售价情形的一个通常的说法是“只要市场能够承受就行”。另一方面，即使一个非常优秀的工程设计也会失去用户的继续青睐，以至最终从市场消失；在这个情形下，遇到了一个不希望的结果，即获利<0。

另一种复杂的情形是 C_{sell}趋向于随时间下降。甚至最热门的新微处理器最终也会变成廉价货。只要改善工程成本投资，这不是一个大问题。复杂的 VLSI 芯片很难设计，而且最初设计花费很大。一个有利的因素是随时间加长

$$C_{chip} \rightarrow C_{materials} \tag{4.10}$$

式中 $C_{materials}$是材料成本。硅的优点是非常便宜，特别是与其他材料，如砷化镓(GaAs)相比时。使一条生产线持续好几年运行可以大大提高它的获利。

这一简短介绍是为了帮助有抱负的 VLSI 设计者理解工业界的整个结构。生产一个含一亿个晶体管的芯片比开办一个“·com”的网站要复杂得多。它要求财政上的保证，很强的技术支持，创造性的工程工作，以及可靠的销售队伍。由于制造过程被认为在获利分析中最花费资金的环节，因此在本章中较为详细地研究它。设计与工程成本则几乎一样高，它们将在本书的其余部分讨论。

本章概要

一个硅芯片可以看成一组做成图案的材料层。当这些材料层被正确地叠放时，所形成的三维结构即是可控开关(晶体管)，它们通过导线相连以实现逻辑操作。

在本章中，首先考察在硅工艺中使用的最重要的材料层，包括氧化层、掺杂的硅区以及金属层。将介绍几种化学反应并简略说明这些材料层在实际上是如何生长或淀积的。然后，进一步研究一个材料层在实际上如何做成图案，使它具有形成导线和晶体管所需要的形状和尺寸。这可以逐步了解制造一个基本 CMOS 电路所采用的步骤。

这一介绍的主要目的是了解最基本的内容以及它们与 VLSI 电路物理设计的关系。许多具有很强科学知识背景的人已对硅工艺表现出浓厚的兴趣，且以这一领域建立起自己的毕生事业。

4.2 材料生长与淀积

一个集成电路是通过把各种材料层按预先设定的顺序堆积而成的。无论材料的电气特性还是材料层的几何形状，对于实现器件和电路的特性都是很重要的。

大多数材料层都是首先形成，然后按照下一节中描述的光刻顺序形成图形。硅掺杂层则与此不同，它们利用光刻工艺定义掺杂物在何处进入硅，从而形成所希望的形状。在本节中将考察在硅 VLSI 工艺中采用的一些基本的工艺步骤。

4.2.1 二氧化硅

二氧化硅(SiO_2)是工艺中极端重要的材料，因为

- 它是一种极好的电绝缘体
- 它能很好地附着在大多数材料上
- 它可以生长在硅圆片上或淀积在硅圆片上面

二氧化硅(SiO_2)一般称为石英玻璃，或简单地称为“玻璃”。除了许多其他的应用外，它也被用做 MOSFET 的栅氧。

在 VLSI 电路中有两类二氧化硅层，它们的差别在于形成的过程不同。**热氧化层**是由以下反应形成的：

$$Si + O_2 \rightarrow SiO_2 \tag{4.11}$$

它利用热作为催化剂。热氧化层的独特之处在于上述反应需要的硅(Si)来自硅圆片本身。如图 4.2(a)所示，氧分子 O_2 通过圆片表面发生反应。这确实是在“生长”玻璃层，其结果如图 4.2(b)所示。图中最终的氧化层厚度用 x_{ox}表示，它取决于温度、晶体取向以及生长时间。由于圆片表面的硅原子参与此反应，因此消耗的硅层厚度为：

$$x_{Si} \approx 0.46\ x_{ox} \tag{4.12}$$

一个含义相同的(也是很有用)的观点是硅表面从它原先的位置上“缩进”。

尽管纯氧产生高质量的氧化层，但这一过程相对较慢。采用蒸气形态的水(H_2O)通过以下反应可以达到较快的生长速率：

$$Si + 2H_2O \rightarrow SiO_2 + 2H_2 \tag{4.13}$$

这称为“湿氧化”。实际中采用 O_2 和蒸气的混合物，并采用作为载体气体的氮及其他化学物

质如氯(Cl)。

热氧层是**自然氧化层**，也就是它是表面暴露在一个氧化气体环境下形成的。如果将一个裸硅圆片放在空气中，就会形成很薄的自然氧化层。提高温度可以提高生长速率。硅氧化的温度一般在大约 850℃～1100℃ 的范围内。

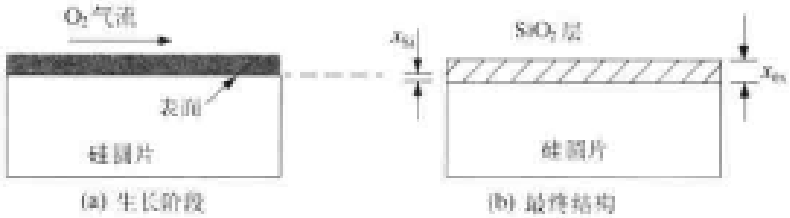


图 4.2　热氧化层生长

在 VLSI 电路中大多数氧化层已处于圆片表面之上，因而没有硅可用于热氧生长。在这种情形中，可以利用气体反应产生二氧化硅分子，然后把它们**淀积**到表面上以提供表面氧化层。这一过程显示在图 4.3 中。采用 SiH_4 的化学反应如下：

$$SiH_4(\text{气体}) + 2O_2(\text{气体}) \rightarrow SiO_2(\text{固体}) + 2H_2O(\text{气体}) \tag{4.14}$$

它可以用来在圆片上产生 SiO_2 的分子。这一技术称为**化学气相淀积**(CVD)，而所生成的氧化层常称为 **CVD 氧化层**。氧化层的厚度是通过生长速率和淀积时间来控制的。也有可能在低温下形成淀积，称为 **LTO**(低温氧化物)。另外，对玻璃进行掺杂常常是有好处的。例如掺磷可以产生"p 型掺杂玻璃"，它对一定类型的平面化步骤有好处。

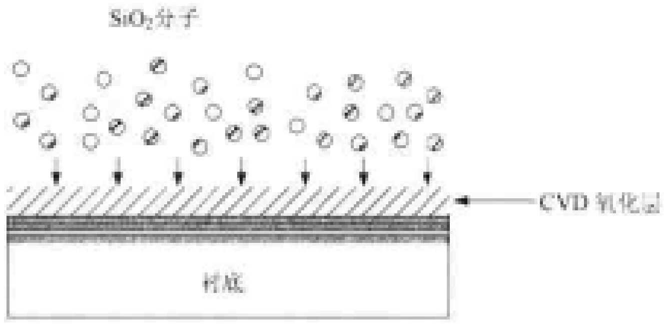


图 4.3　化学气相淀积(CVD)氧化工艺

4.2.2　氮化硅

另一种有用的材料是氮化硅 SiN_4，它在上下文清楚时也常常就称为氮化物。以下的反应则是生成氮化硅的一种技术：

$$3SiH_4(\text{气体}) + 4NH_3(\text{气体}) \rightarrow Si_3N_4(\text{固体}) + 12H_2(\text{气体}) \tag{4.15}$$

氮化硅的特点是它对大多数的原子可以起到阻挡的作用，这使它作为玻璃表面覆盖层时非常理想，玻璃表面覆盖层是芯片的最后一层保护覆盖层，因为它可以防止污染物接触易受影响的硅电路。在制造过程中，氮化硅用来在电气上隔离相邻的场效应管(以后会讨论)，而且它具有

较大的介电常数 $\epsilon_N \approx 7.8\epsilon_0$,因此常考虑在许多电容结构,如 DRAM(动态随机存取存储器)单元中用做起隔离作用的 ON(氧化物-氮化物)绝缘夹层。

4.2.3 多晶硅

如果在非晶 SiO_2 层上淀积硅原子,那么硅就会结晶,但却找不到与之相应的可以参照的典型晶体结构。它们形成小的**晶体**,即为硅晶体的小区域。这样的材料因此称为**多晶体硅**或**多晶硅**,或简称"**多晶**"。多晶硅普遍用做场效应管的栅材料,它具有一些我们十分希望有的特性,即它可以被掺杂,能与氧化硅结合得很好,并且可被覆盖一层高熔点(耐熔)的金属,如钛或铂,以减少薄层电阻。在 CMOS 集成电路中多晶为形成 MOSFET 提供非常好的基础。

采用氮化硅淀积硅原子的基本反应为:

$$SiH_4 \rightarrow Si + 2H_2 \tag{4.16}$$

该反应发生在温度大约为 500 ℃~600 ℃ 近年来在制造现代动态随机存储器(DRAM)单元的**堆叠电容**时,多晶淀积技术得到了发展与提高。这些将在第 13 章的 13.3 节中研究。

4.2.4 金属化

铝(Al)是最常用的集成电路互连线金属。它可通过在真空腔中加热蒸发,所形成的蒸铝流体用来覆盖圆片。铝有很好的粘附特性并且很容易做成图案形状,因此它获得普遍使用是不难理解的。

铝的体电阻率约为 $\rho = 2.65\ \mu\Omega\cdot cm$。一条厚度为 0.1 μm 的铝互连线其薄层电阻约为:

$$R_s = \frac{\rho}{t} = \frac{2.56\times10^{-6}}{10^{-5}} = 0.265\ \Omega \tag{4.17}$$

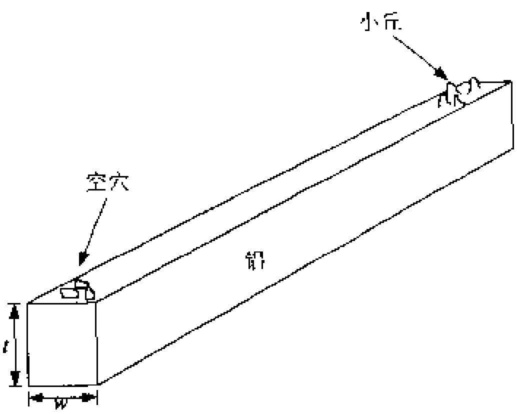


图 4.4 铝的电迁移效应

但铝存在一个称为"**电迁移**"的问题。即高电流密度确实会使原子从一条互连线的一端移出,从而形成称为"**空穴**"的小坑。在另一端堆积起来的原子形成称为**小丘**的微结构,如图 4.4 所示。小丘和空穴会引起导线失效,因此针对这一问题已展开了许多研究。一个通常的解决办法是在金属淀积阶段使铜和铝混合。这将减少电迁移的影响,但却使电阻率增加到大约 3.5 $\mu\Omega\cdot cm$,薄层电阻也将按比例增加。

在物理设计层次,通过控制在互连线中的电流密度 $J(A/cm^2)$,避免过多地形成小丘和空穴。对于厚度为 t 和宽度为 w 的互连线电流密度为

$$J = \frac{I}{A} \tag{4.18}$$

式中 I 为电流,单位为安培,而 $A = wt$ 为横截面积,单位为平方厘米。版图设计者不能改变材料层的厚度 t,因为厚度是由工艺线确定的。因此电迁移是通过确定使 J 低于最大的电流

密度值J_{max}所要求的最小线宽 w 来控制的。这是我们遇到的第一个版图**设计规则**的例子，它说明了对于一种特定情形所允许的最小特征尺寸。在本章的后几节中将比较全面地考察设计规则。

MOS 起始于金属栅工艺，字母 M 就是代表金属，而铝曾被选做栅极层。管子栅极采用铝的缺点是铝的熔点低，一旦它被淀积在圆片上就不能再进行高温工艺步骤。随着工艺技术不断完善，工艺工序日益复杂，铝栅已成为一个限制因素。于是就开发多晶栅的管子，现在已是 CMOS 的标准工艺。采用硅栅的一个较大的问题是即使重掺杂的多晶也具有较大的薄层电阻，其值约为 $R_s=25\sim50\ \Omega$。为了克服这个问题，多晶上覆盖一层**难熔**（高温）金属薄层，如钛（Ti）、钨（W）或铂（Pt）。这个多晶－金属的组合称为**硅化物**，但这一组合在设计中通常都被看成单个层。这在后面要介绍的 CMOS 工艺工序中将会明白地显示出来。钨也常用来作为连接金属层间的通孔的插塞。

近来，已采用铜（Cu）替代铝。因为铜的电阻率几乎是铝的一半，所以它的薄层电阻较小。在器件层次上，这一差别并不重要。然而，当铜用来作为很长的系统一级的互连线时，减少薄层电阻是非常有意义的。工艺改进的实现并不容易，标准的图案形成技术并不能用在铜材料上，因此必须开发特殊的工艺技术。在 4.4 节中将讨论采用铜的问题。

4.2.5 掺杂硅层

硅圆片是 CMOS 制造过程的起始点。它在晶体生长期间被定义为 n 型或 p 型，且是整个电路结构的衬底基础。按照定义，一个掺杂的硅层是圆片表面形成图案的 n 型或 p 型部分。尽管硅层并不总是按照通常意义堆叠起来，还是采用这一说法以保持一致。

在衬底上形成掺杂层的关键是把施主或受主原子引入圆片中，这些原子最终将结合到硅晶体中。在现代 CMOS 中，这是由称为**离子注入**的技术实现的，即原子先在一个腔室中被电离，然后在粒子加速器中被加速到很高的能量。离子束通过一个质量分离单元，该单元采用磁场来选择所希望的电荷种类。整个系统显示在图 4.5 中。快速运动的离子几乎以典型值约 100～200 keV 的能量被击入衬底中。离子在与硅圆片中的电子和原子核碰撞几次后便静止下来，如图 4.6 所示。这个减缓下来的过程破坏了晶体且使掺杂剂留在随机的位置上。为了恢复晶体且使掺杂剂处于晶格内的合适位置，圆片需在**退火**步骤中被加热。在退火过程中掺杂剂会由于称为**粒子扩散**的过程而稍稍重新分布；扩散只不过是集中在一个小区域内的粒子由热引起的集体运动，这一运动使粒子散布开来。

离子进入硅中的分布可以采用高斯分布形式进行一阶近似：

$$N_{ion}(x)=N_p e^{-\frac{1}{2}\left(\frac{x-R_p}{\Delta R_p}\right)^2} \tag{4.19}$$

其单位为 cm^{-3}，圆片的表面处定义为 $x=0$。这一函数显示在图 4.7 中，图中 R_p 称为**投射范围**（projected range），它是注入离子的平均深度。R_p 的值取决于入射能量、种类、晶体取向，其范围可从大约 0.1 μm 到 1 μm 深。峰值密度 N_p 发生在 $x=R_p$ 处。方差用**分散程度** ΔR_p 来表示，它表示由于能量损失过程的统计特性各个离子在停止深度上的差别。更为精确的注入剖面形态模型可以采用 Pearson 类型 IV 的分布及数值模拟。

注入离子的数目通常用注入剂量 D_I 表示，其定义为：

$$D_I = \int_{全部x} N_{ion}(x)\,dx \qquad (4.20)$$

其单位为每 cm^2 的离子数(或即 cm^{-2}),它可以用电荷计数器非常精确地测量。当分析 MOS 电容的宏观电特性时常要用到这个注入剂量。

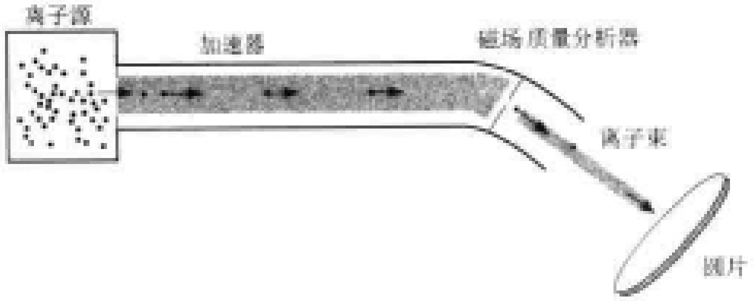


图 4.5 离子注入器的主要部分

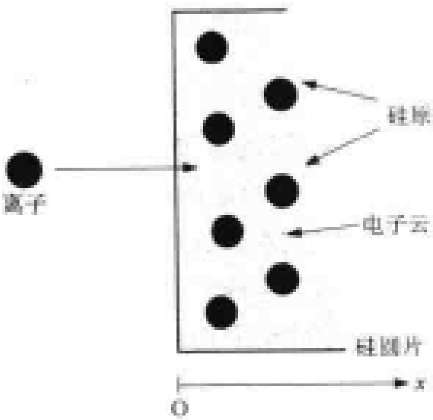


图 4.6 离子停止的过程

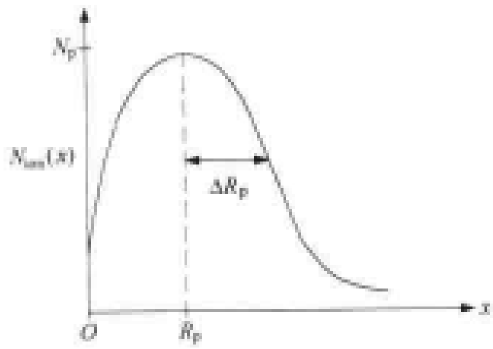


图 4.7 高斯分布的注入剖面形态

4.2.6 化学机械抛光

假设淀积多晶硅并且形成图案,然后在它上面淀积二氧化硅。那么如图 4.8(a)所示,被淀积的二氧化硅表面就会有一个"小丘",这是由于它下面的多晶硅线所致。如果在二氧化硅上再淀积一层金属互连线,它就会跟随这个表面的轮廓起伏,但由于这一变化它必定更宽和更厚一些。如果继续增加金属层次,那么层面就将越来越凹凸不平,并且可能引起细线条的断裂和其他问题。当只采用一层或两层金属互连线时,未平面化的表面不一定是个问题。但在现代 CMOS 工艺中,五层或更多层的互连线十分普遍,因此使表面平面化的技术已成为必不可少。

化学机械抛光(CMP)采用化学刻蚀和机械"喷沙"在硅圆片上产生平整的表面。当它应用到氧化物上时就会形成一个很平的表面如图 4.8(b)所示。CMP 步骤选用在 CMOS 制造中,当具有一个平整的工作表面是非常重要的几道工序处。这几处包括金属淀积步骤以及下

节将要讨论的光刻过程中应用光刻胶的时候。

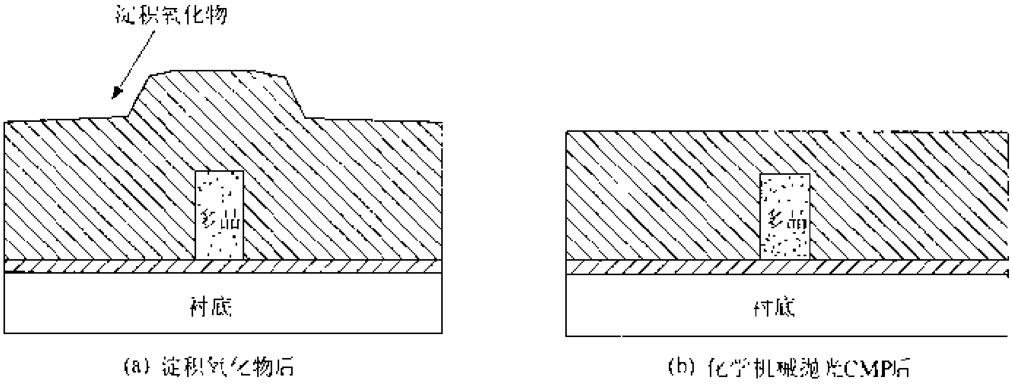


图 4.8 表面平面化

4.3 刻蚀

我们已把集成电路定义为一组三维的具有图案形状的材料层。在现代 CMOS 制造中最关键的问题之一是采用一种技术在每一材料层上形成具有亚微米特征尺寸的图案，这是运用**光刻**工艺来实现的。在光刻工艺中，用光学方法把图案的阴影投射到芯片表面上，然后采用类似于照相的技术把图案转移到它上面。这与制造印刷线路板的工艺相同，但芯片制造允许的分辨率要小于 0.12 μm。光刻已经发展成一门非常复杂的学科，它需要继续缩小特征尺寸。这里的概述足以帮助理解光刻的主要特点以及它与 VLSI 系统设计的关系。

光刻工艺从定义材料层所希望的图案开始。这是一个计算机数据库格式的文件，它在芯片版图设计阶段生成。用这一数据去制造一块高质量的玻璃，其上的图案是用一种金属如铬来形成的。这就是所谓的掩模(reticle 或 mask)，它通常为实际芯片的 5～10 倍。因此掩模包含两个区域：透明(没有金属)的和不透明(有金属)的区域。图 4.9 为一掩模的构成。当光照亮掩模时，它就把掩模图案的阴影投射到芯片的表面。

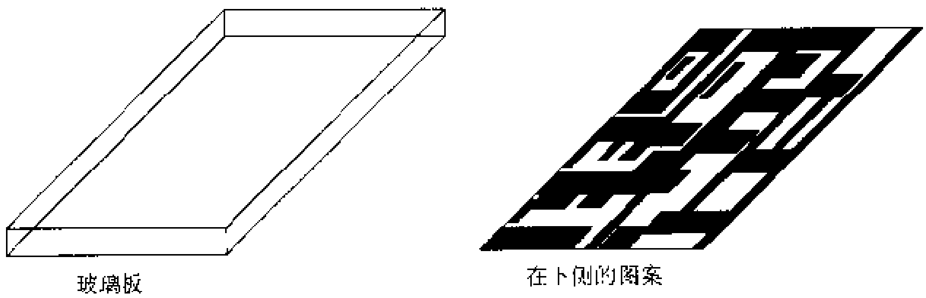


图 4.9 掩模是一块其上有铬图案的玻璃板

为了把掩模图案转移到一个硅区表面，先在圆片上涂一层光敏液态塑性材料称为**光刻胶**(或简称为**胶**)，这一过程表示在图 4.10 中。图 4.10 (a)表示液态光刻胶正在被喷撒到用真空吸盘定位的旋转圆片上。使圆片旋转可以利用离心力把光刻胶涂满整个表面，从而形成相当均匀的涂层如图 4.10 (b)所示。但在圆片周边附近有所不同，由于表面张力引起**鸟嘴效应**，如

图 4.10 (c)所示，这使圆片的可用区域被限制在周边以里的内部区域。

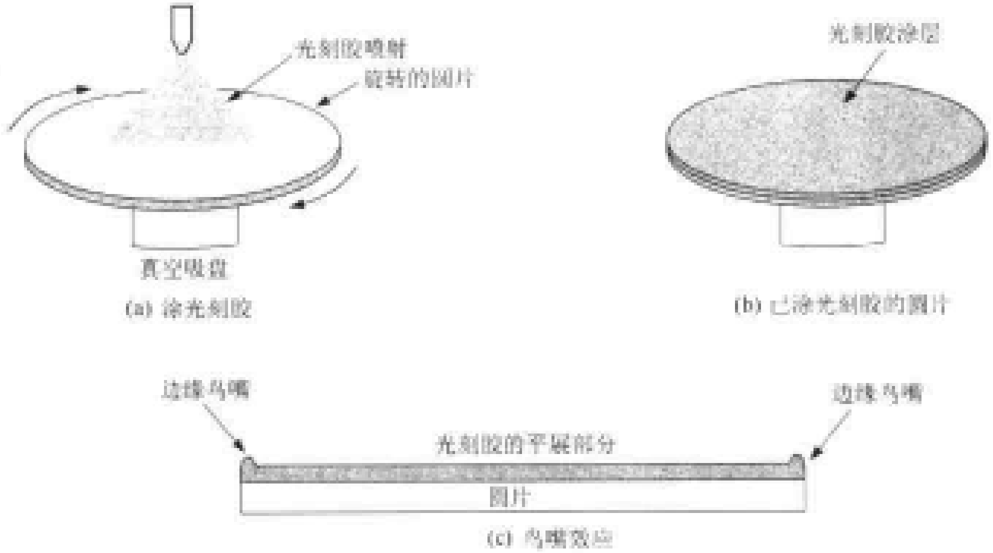


图 4.10　涂光刻胶

光刻胶非常类似于照相底片，即它对光很敏感。VLSI 光刻胶能对光谱的紫外线(UV)区域产生反应，在这一区域光子的能量最高而波长最短。这就是光刻工艺中的**曝光步骤**，图 4.11 说明了它的主要概念。在曝光完成后，光刻胶就用化学冲洗(rinse)方法显影。大多数的 VLSI 工艺采用**正光刻胶**(正胶)。所谓正胶工艺就是光被遮住的区域在冲洗过程中变硬，而曝光的区域则被冲洗掉。正胶的特点显示在图 4.12 中。图 4.12(a)中的曝光步骤确定了在掩模投影中的亮区和暗区。光刻胶显影后，在被遮光的区域保留变硬的部分，这显示在图 4.12(b)中。负光刻胶则具有相反的特点：光照的区域变硬，而遮光的区域则被溶解和冲洗掉。

变硬的光刻胶层用来保护它下面的区域免受**刻蚀**工艺的影响。所谓刻蚀就是使圆片的表面接触由惰性气体如氩(Ar)形成的气体等离子体，其中含有反应化合物；这整个过程称为**反应离子刻蚀**(RIE)。所选择的化合物和等离子体用来刻蚀掉没有被硬化光刻胶保护的材料层。在刻蚀过程中，硬化的光刻胶本身是能经受住刻蚀剂混合物的。图 4.13 显示了一个刻蚀的例子。在图 4.13(a)中，氧化层顶部形成了一个光刻胶的图案，刻蚀步骤移去了未被保护的区域的氧化物，因此氧化物具有与光刻胶相同的图案，如图 4.13(b)所示。这一技术可以使圆片表面上的任何材料层形成图案，包括多晶硅、CVD 氧化物以及金属，① 它能把计算机版图设计图案转移到实际的硅层上，完成一个逻辑电路的物理实现。

掺杂硅区也是采用光刻工艺来形成图案的，但是顺序不同。此时，在圆片上生长氧化层，然后采用光刻向下刻蚀到硅表面，这与图 4.13(b)所示横截面的情形是完全一样的。然后光刻胶－氧化层用来遮盖硅免受离子注入的影响。图 4.14(a)表示砷离子束入射整个表面，但这一掺杂剂只能接触到氧化物已被刻蚀区域的硅。因此所生成的 n+ 图案是由氧化层的开口确定的。注意，n+ 图案比氧化层开口稍微宽些，这是由于受所谓**横向掺杂**的影响，它是由于在退火阶段掺杂剂的扩散所致。横向效应会限制窄线条光刻系统的分辨率。

---

① 铜是一个例外，因为它是用不同的技术来形成图案的。

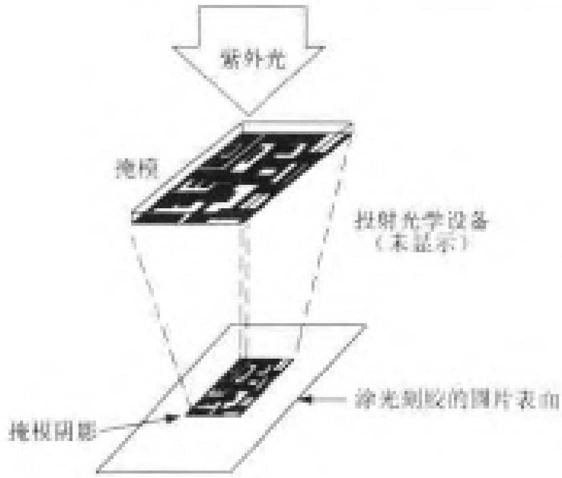


图 4.11 曝光步骤

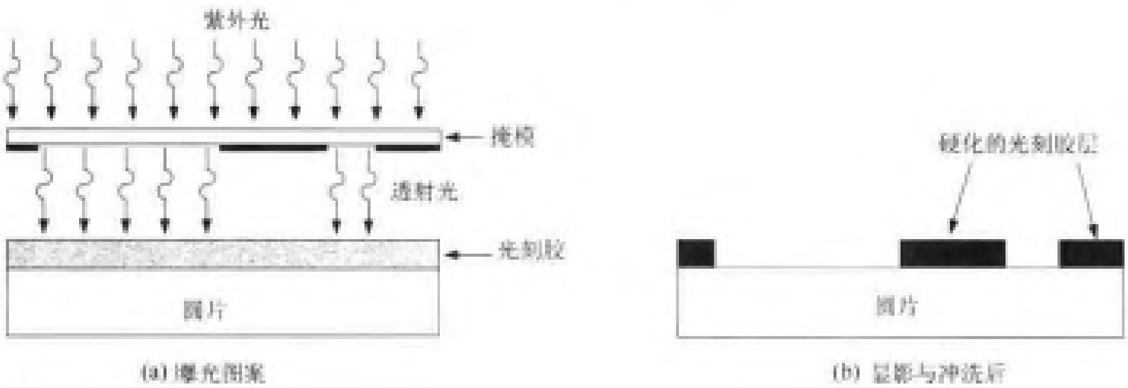


图 4.12 正光刻胶的特点

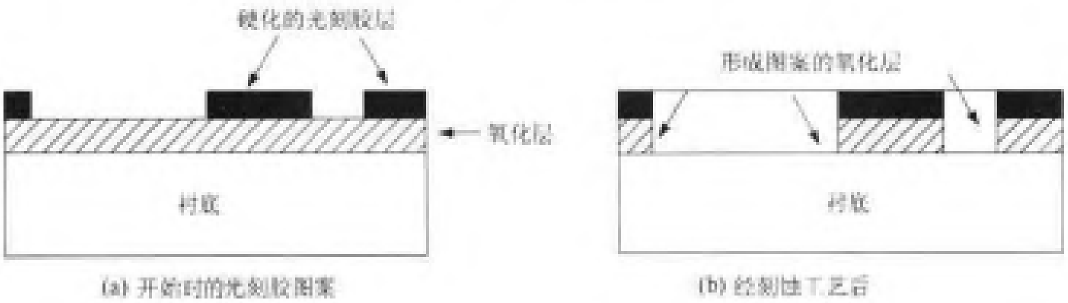
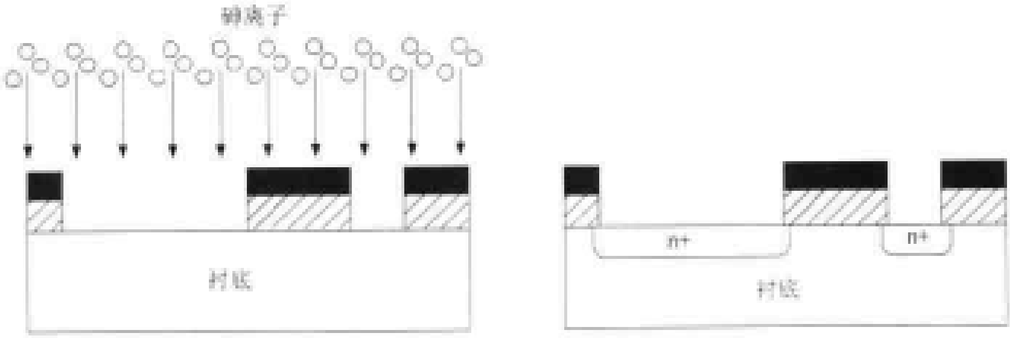


图 4.13 氧化层刻蚀



(a) 注入的离子束 (b) 掺杂的n-型区

图 4.14 形成掺杂硅层图案

这里显示的只是一个图案,但制造过程中采用的是含有许多芯片部位的大圆片,每个芯片部位用"**步进和重复过程**"一个一个地曝光。一个**光刻机**用来固定圆片的设备,它可以精确地移动,使每次光照与一个芯片部位对准。在一个芯片部位被曝光后,光刻机就把圆片步进到下一个芯片部位。这个工序使圆片上形成了大量相同的芯片部位,如图 4.15 所示。**测试部位**处含有各种测试结构与电路,如 MOS 电容,掺杂硅区,MOSFET 以及简单的电路。这些测试结构与电路用来在制造工序的不同阶段对圆片进行电气测试。**圆片探针**是一组非常小的金属探针,它们可以接触圆片上的区域以进行测试。从探针读出的数据可以提供有关制造流程是否进行得很好的信息,且能提供电路设计所需要的关键的电气参数数据。在圆片上包含几个测试部位是非常重要的,它们代表了圆片上所有的区域,因为不均匀的温度、气体流的密度以及其他参数在圆片各处不同,这会影响电气特性。

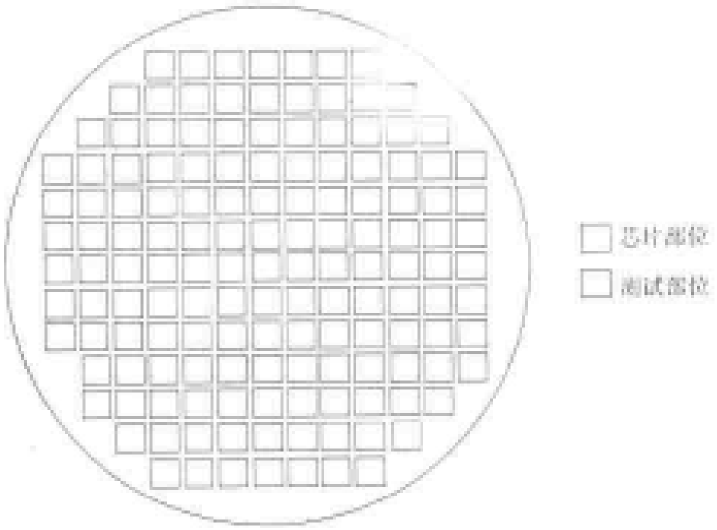


图 4.15 圆片上的芯片部位

光刻工序针对制造集成电路所需要的每层掩模步骤重复进行。值得注意的是,第 1 层掩模步骤确定了芯片图案的基本轮廓;后续的掩模步骤必须使形成的图案相对已经在衬底上形成的图案具有正确的间隔距离。使一层掩模与其他层掩模图案正确对准,对于成品率十分关键,掩模没有对准可能会使整个芯片不工作。精确的对准可以采用"**套准标记**",它是在基础层

上形成的几何图形，只是用来帮助与后续的掩模步骤对准。随着各层的形成，将需要用到更多组的套准标记。

洁净间

光刻工艺对于灰尘颗粒是非常敏感的。如果一小颗灰尘落在光刻胶上，它将影响曝光和显影，可能产生一个缺陷。类似地，如果一个灰尘颗粒掉在光照聚焦平面范围内的掩模上，它就会把颗粒的形状映射到圆片的芯片部位上。像这样一些情形都会降低成品率，这在亚微米几何尺寸的情形下尤为严重。

已经开发许多方法来解决这些问题。光刻是在**洁净间**的环境中完成的，它采用 HEPA（高效颗粒气体）过滤器去除灰尘颗粒，HEPA 过滤器必须能有 99.7% 的效率去除直径大于 0.5 μm 的颗粒。一个 **X 级**的洁净间是指每立方英尺中直径大于 0.5 μm 的颗粒应当少于 X 个；在关键工作区的现代设备具有 1 级或更高的级别。为了保证这一洁净程度，工人们在进入工作区前必须进行空气浴并穿上能覆盖身体所有部位的特殊服装；这些服装由于它们的外观而常常被称为"兔装"。另一个办法是整个流程都自动进行，所有的移动均由机器人来完成。

光刻区采用黄光照明，因为它不会影响对紫外线敏感的光刻胶。为了防止灰尘颗粒落到掩模上破坏图案，在掩模上方放置一层很薄的透明塑料收集灰尘使之远离掩模表面。这称之为**掩模罩**（pellicle），它放置在掩模之上足够的高度使灰尘不会出现在投射光的成像平面上。

工艺环境还有许多其他特点，以保证生产出功能正确的芯片。需要有许多科学家、工程师和技术员来设计，维护和更新工艺操作区。参观一个现代化芯片制造厂对了解 VLSI 技术是极为重要的。

4.4　CMOS 工艺流程

现代 CMOS 工艺无论从哪个角度看都是一种"技术上的奇迹"。CMOS 生产线几乎是从"沙"开始，生产出微小的矩形片子，它们能为全世界提供强大的计算功能。许多半导体制造公司开发出高度先进的工艺技术，但它们的工艺流程细节则是各家高度专有的。由于建设一个新的制造厂需要十几亿美元，因此各家公司必须保持高度机密是不足为怪的。

在本节中，将研究一个"标准"硅 CMOS 工艺的主要步骤。介绍的层次选择在确保要点而不涉及过多的细节上。了解 CMOS 工艺对于每个 VLSI 设计者来说是非常重要的，它对某些设计者来说甚至更为重要，这取决于各个工程师正在从事的任务。器件和电路工程师把工艺参数看成对于他们的管子和电路究竟能翻转得多快的基本限制。系统结构设计者懂得逻辑功能块是需要制造在硅上的，因而工艺决定了面积的分配、互连线的层数、延时、时钟速度，以及许多其他系统层次上的考虑。涉及 VLSI 芯片设计的每个人都会受到工艺的影响。

图 4.16 为开始阶段的几个步骤。应当注意图中的一些尺寸，特别是在垂直方向上的尺寸并没有按比例画，否则无法显示一些重要的细节。图 4.16(a)表示起始于一个 p+ 的圆片，在它上面生长 p 型的硅**外延层**。外延层通过把硅原子投入被加热的圆片上、形成制造管子所需要的高质量晶体层。圆片本身作为制造芯片的衬底，在其余的图中它将不明显画出。

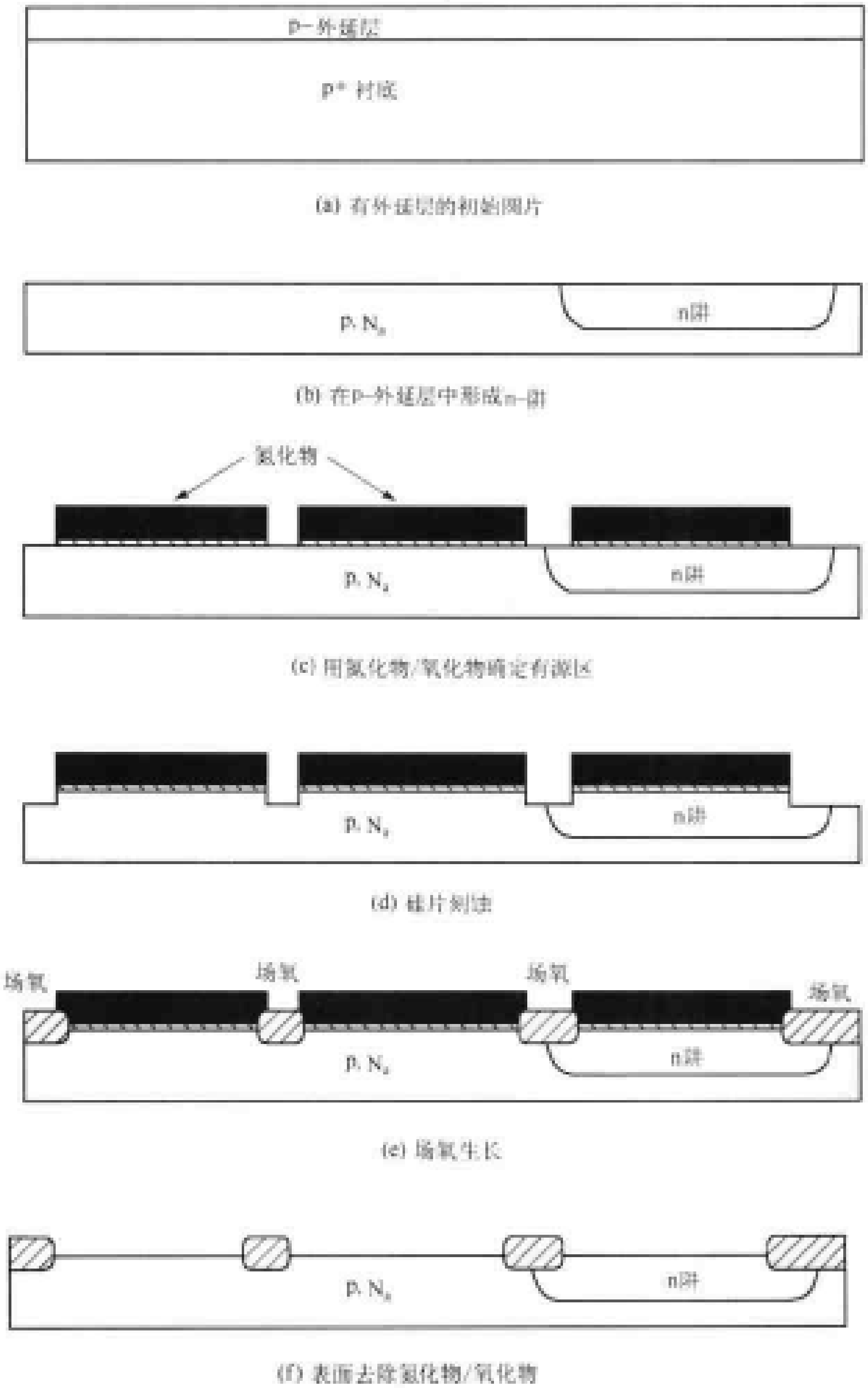


图 4.16　CMOS 制造中起始的几道工序

图 4.16 (b)表示下一个步骤是运用掩模步骤来形成 n 阱区域，它确定 pFET 管的位置。一般地，每一个管子(nFET 或 pFET)是在圆片表面的**有源区**形成的。有源区是通过掩模步骤来确定的，它使在薄热氧化层上的氮化硅形成图案，而这层氮化硅则用来释放晶体表面的机械应力。图 4.16 (c)为图案形成后的细节图。有源区的形成也是**电气绝缘**技术的一部分。这个技术采用凹下去的玻璃(氧化物)区域作为隔离体，阻止相邻器件之间发生电气导通。为了达

到隔离，运用氮化物图案确定刻蚀掉的硅区，如图 4.16 (d)所示。在已刻蚀区生长或淀积氧化物，如图 4.16 (e)所示。在有源区之间的玻璃绝缘形成了**场区**，因此在那里的氧化物称为**场氧**或 **FOX**。一旦生长了 FOX，氮化硅－氧化层就被去除以露出硅表面。于是由图 4.16 (f)横截面图所示的圆片，可以用于管子的制造过程。

FET 的形成采用**自对准栅工艺**。在这个技术中，首先形成栅，然后把它作为离子注入的掩模，形成 n+ 或 p+ 漏/源区。起始点是生长栅氧，如图 4.17(a)所示。在这一步骤中确定栅氧的厚度 t_{ox}，然后淀积多晶硅并且做成图案以形成晶体管的栅极。所形成的图 4.17(b)的结构表示到这步骤时的横截面图。为了形成管子，需要在硅上形成掺杂的漏区和源区。pFET 的形成采用**pSelect**(选择p的)掩模图案并用硼离子注入。如图4.17(c)所示，选择p的掩模

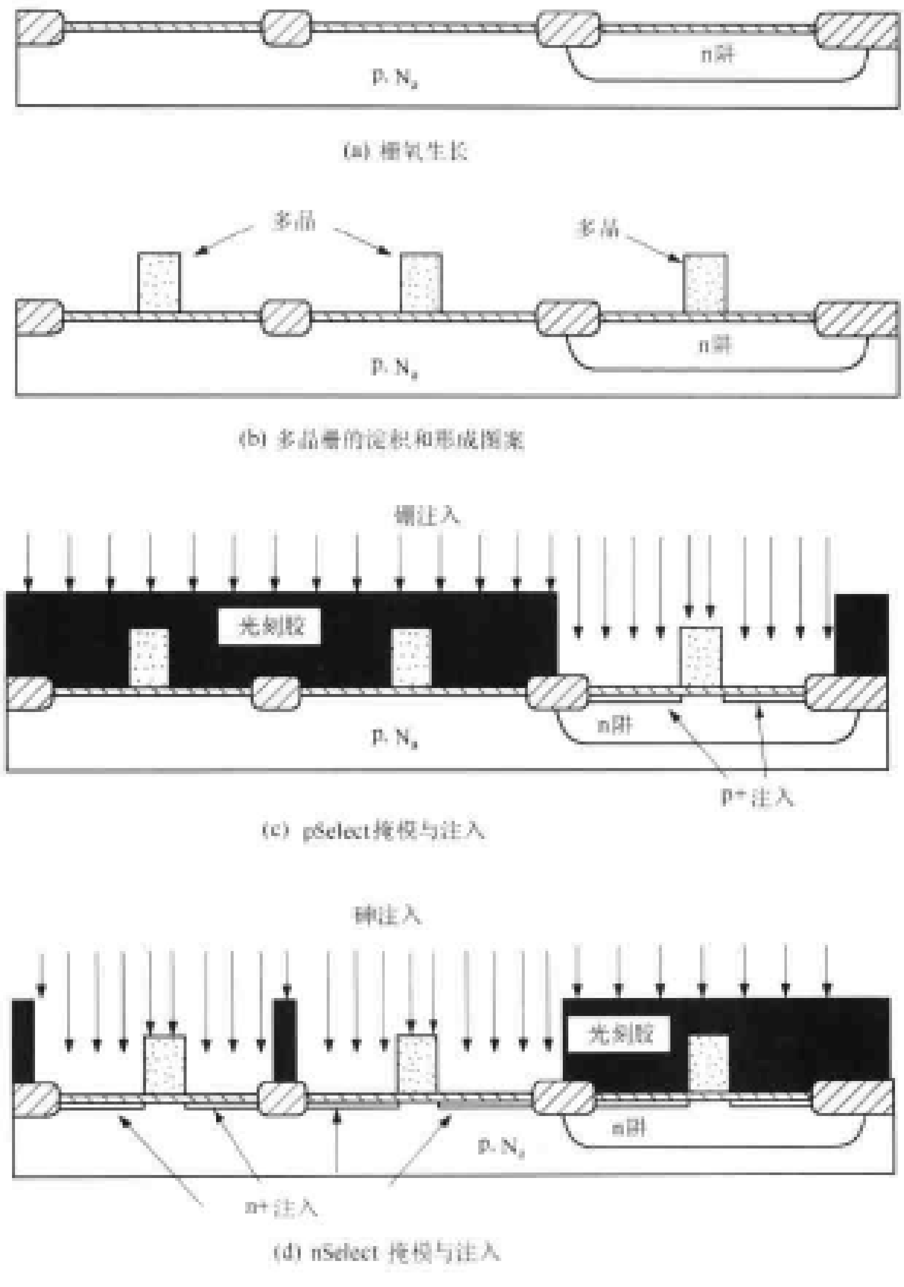
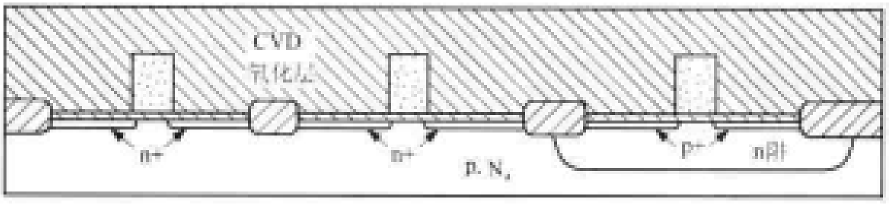


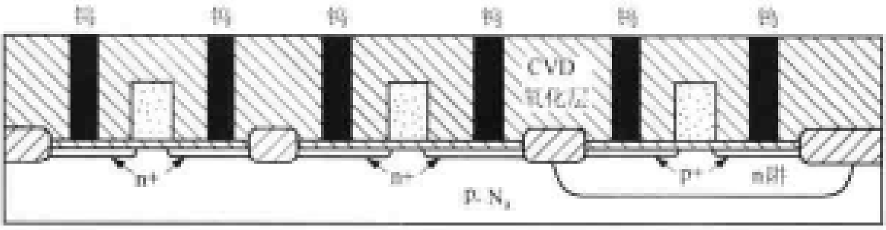
图 4.17　nFET 和 pFET 的形成

所形成的硬化光刻胶层阻挡在 nFET 位置上的离子注入，但却允许离子束进入 pFET 区域。离子被栅多晶层吸收，但很容易使它穿过薄氧化层达到硅层。"自对准栅"的名称由此而来。nFET 也以类似的方式形成。一个 **nSelect** 掩模用来阻挡 n 型离子注入 pFET 区，离子只允许轰击 nFET 区以形成 n+ 区，如图 4.17(d)所示。到此，所有的管子都已形成。硅化物栅可以通过在多晶栅上加一层难熔金属来形成，它可以降低多晶线的薄层电阻。工艺流程中的其余步骤用来形成互连线层。

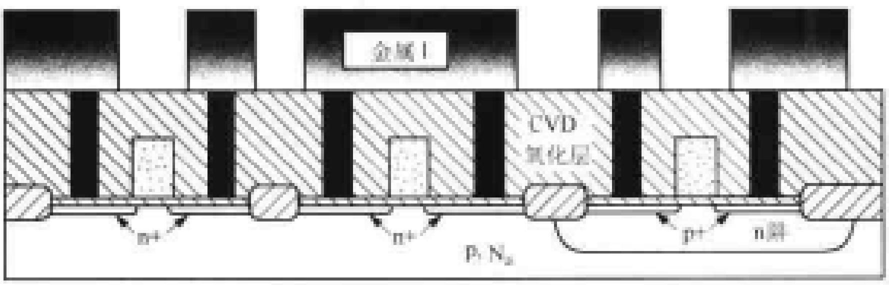
增加互连线层的基本顺序如图 4.18 所示，图中为第一层金属。采用 CVD 氧化物覆盖表面如图 4.18(a)所示。引出 n+ 和 p+ 区域的电连接接触孔是通过**有源接触孔**掩模在氧化物上刻蚀孔形成的。在刻孔完成后，这些孔中就填充以金属塞的材料如钨(W)。所形成的结构如图 4.18(b)所示。然后淀积第一层金属且用 Metal 1 掩模使它形成图案。这个掩模确定了把电路连接在一起的第一层金属互连线。图 4.18(c)所示为在第一层金属形成图案后的最终剖面图。可用同样的方式加入更多的金属层。当前的工艺线已有 5 层或 5 层以上的金属互连层(它们间由氧化物隔离)以实现复杂的布线。



(a) 退火和 CVD 氧化层



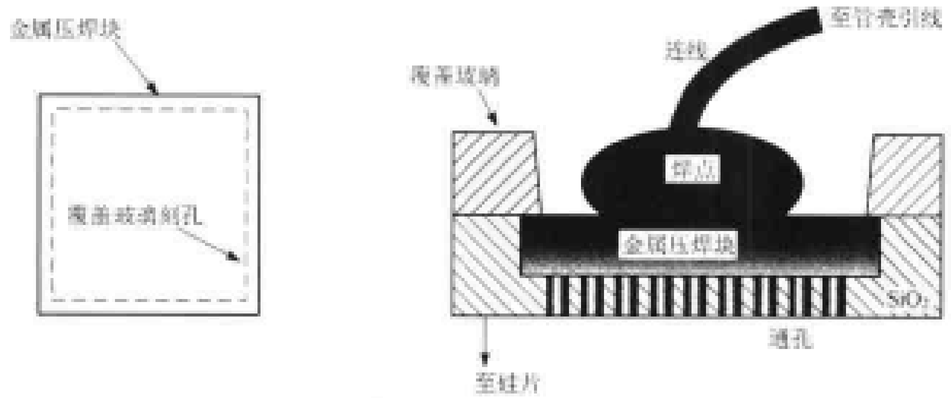
(b) CVD 氧化层中有源区接触及钨塞形成之后



(c) 金属1 涂层及图案形成

图 4.18 第一层金属互连层

在所有的金属层形成之后，整个芯片敷以覆盖玻璃以保护表面免受外部污染。氮化硅是最常用的覆盖玻璃材料，因为它是致密的电介质，可以阻止不希望有的原子的扩散且能与金属结合得很好。由于它是绝缘体，所以必须刻蚀出孔，与芯片间实现电连接，这需要另一层掩模。使硅电路与外界接口的最简单的方式是布置**压焊块结构**，即使大的金属**压焊块**排布在芯片核心区域的周围。在压焊块和封装上的输出引线之间用导线连接。图 4.19 显示了这个基本概念。图 4.19 (a)的顶视图显示了金属压焊块(实线)和覆盖玻璃刻孔(虚线)。压焊块本身也许是很大的，在某些工艺中采用的压焊块为 100 μm × 100 μm。图 4.19 (b)的侧视图显示了焊接本身的细节。一个机械手装置把焊点精确地放在压焊块上，且把导线从芯片引到封装管壳上指定的引线上。



(a) 顶视图　　(b) 侧视图

图 4.19　压焊块结构

工艺改进

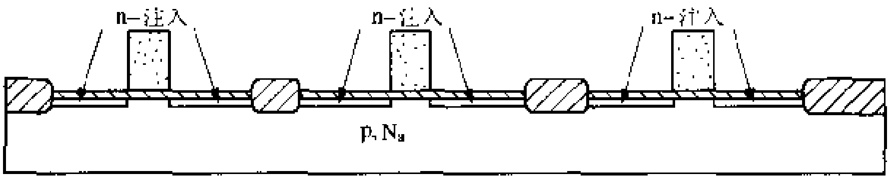
现代 CMOS 工艺线对以上介绍的基本流程进行了许多方面的加强，通常包括提供较好的电气特性，克服小器件或高密度带来的问题，或者提高成品率。我们将考察两个附加的步骤，现在它们已成为标准步骤：即轻掺杂漏区(LDD)FET 以及硅化物。此外，将简述铜互连线图案是如何形成的。

轻掺杂漏区 MOSFET 是设计用来降低在沟道区的电场，这是通过提供 n^-(轻掺杂)的漏区和源区而不是通常的 n+ 区。在理论上这可降低最大的电场强度，从而提高管子的可靠性。[②] LDD 结构的形成可以不需要另外的掩模，因此它们与版图设计者没有太大的关系。

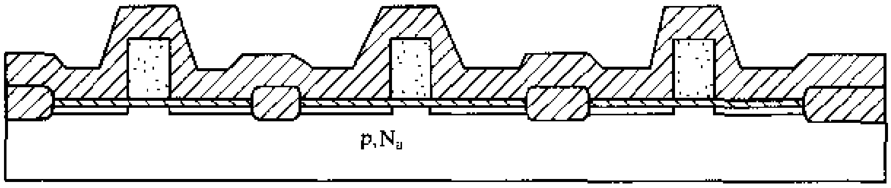
图 4.20 为形成 LDD FET 的顺序图。起始状态表示在图 4.20(a)上。为了形成一个 n 沟道的 MOSFET，开始时采用低剂量的施主掺杂形成 n^-(轻掺杂)的漏区和源区。下一步，如图 4.20(b)所示，是在表面上淀积一层氧化物。注意氧化层也覆盖了多晶栅的侧墙(垂直方向)。之后圆片进行氧化物刻蚀。当从顶部看时，侧墙氧化层要比覆盖平坦表面部分的氧化层来得厚些。这就形成了**侧壁隔离墙**(sidewall spacer)，如图 4.20(c)所示。侧壁隔离墙用来阻挡图 4.20 (d)中所示的重掺杂 n+ 施主的注入，从而保持源区和漏区最接近沟道的地方是轻掺杂。

② 这是由于运用 LDD FET 减少了在短沟道器件中的热电子效应。

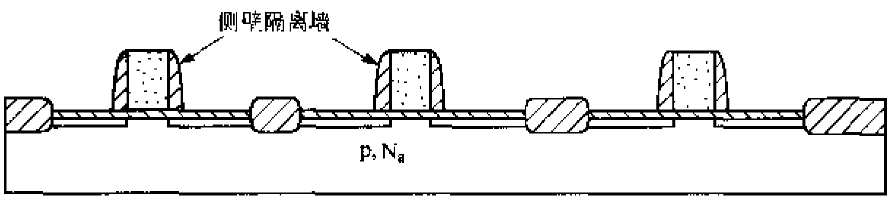
侧墙的横向(水平)宽度决定了 n-区域的延伸范围。



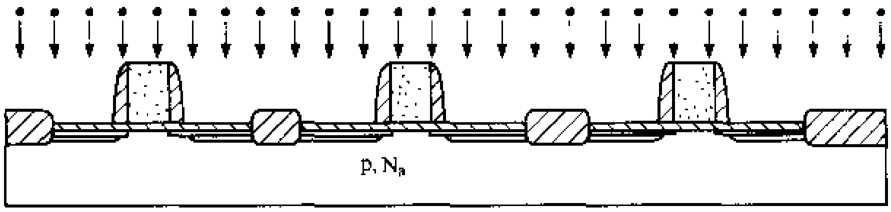
(a) 轻(n-)注入



(b) 覆盖氧化层



(c) 刻蚀后



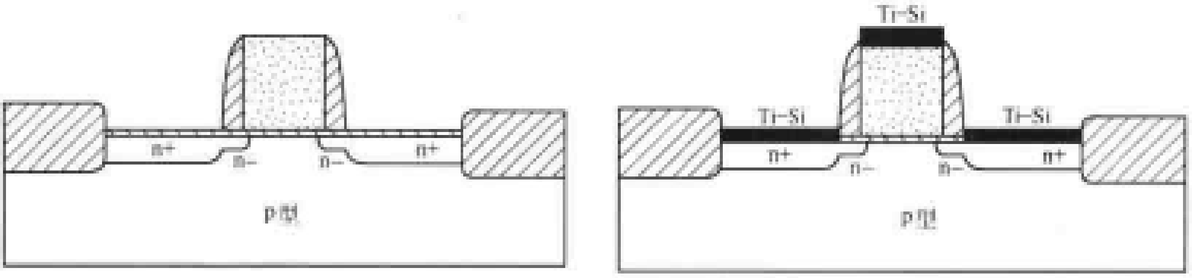
(d) 施上重注入

图 4.20 形成轻掺杂漏区 nFET 的工序

图 4.21(a)所示为完成的 LDD nFET 的放大图,显示掺杂区的细节。它也可以作为研究硅化物的基础,硅化物是将要考察的第二个与 CMOS 基本流程的不同点。即使是重掺杂的多晶硅,其薄层电阻也有大约 25 Ω 或更大,从而限制了它作为互连材料的应用。为了克服这个问题,可以把一种难熔金属,如钛或铂,覆盖在硅或多晶硅上,如图 4.21(b)所示。所形成的硅化物将降低多晶层的薄层电阻,而不会影响 MOS 栅结构的电气特性;硅化物多晶薄层电阻的典型值在 $R_S \approx 10$ mΩ 的数量级上。当采用钨塞作为有源接触时,漏-源 n+ 硅化物可以降低接触电阻。由于这一事实,硅化物在高频工艺中非常普遍。我们注意到,无论是铂还是钨,就它们本身而言都不能用来代替多晶硅栅(以形成真正的 MOS 结构),这是因为它们并不能与二氧化硅绝缘层结合得很好,而只会"滑离"。

要考察的最后一个工艺改进是运用铜代替铝,作为互连线的材料。一个熟知的事实是铜的体电阻率为 $\rho = 1.67\ \mu\Omega\cdot\text{cm}$,它大约为铝的一半。当用来作为互连线材料时,它的薄层电阻大约为具有相同厚度的铝线的一半。然而,铜一直很难引入到工艺线上。它不能采用标准的

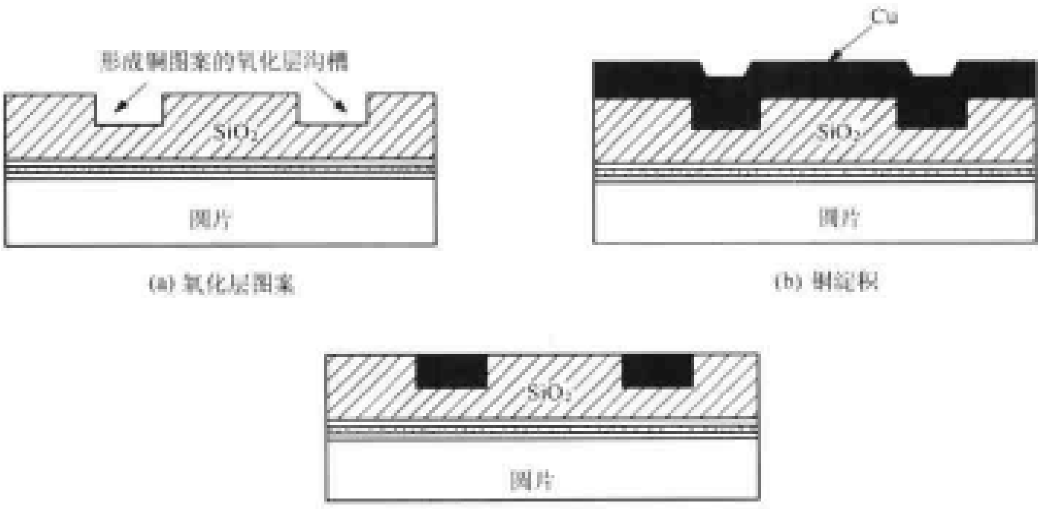
淀积而后光刻的步骤来形成图案，因为采用标准的 RIE 技术很难刻蚀铜。铜能通过硅很快地扩散而改变电特性，因此它不能直接淀积在任何硅区上。它也能通过二氧化硅扩散，这甚至使问题更难解决。曾经进行了许多研究希望开发出能用低电阻率的互连线金属来代替铝的技术。现在，铜正在被引进到大多数新的高速 CMOS 工艺线，VLSI 设计者对此非常有兴趣。第一批运用铜技术的芯片之一是新一代的 Power PC 微处理器设计。



(a) LDD FET结构　　(b) 硅化物的形成

图 4.21　具有硅化物栅和接触孔的 LDD nFET

让我们考察如何形成铜图案的。如上面提到的，干法刻蚀技术不能刻蚀铜。甚至 Al-Cu 混合物中极少量的铜也很难从芯片表面去除。为了解决这个问题，采用大马士革(Damascene)镶嵌工艺，它是基于古代把金或银镶嵌到铁剑中去所用的方法。这一工艺的名字就是出自大马士革城市的名字，这座城市工匠的作品是非常著名的。在这一技术中，铜的图案被刻入到二氧化硅层中，然后把铜淀积(例如运用电镀)到表面上，这一顺序显示在图 4.22(a)和(b)中。为了避免采用刻蚀，将圆片进行化学机械抛光(CMP)，这一步骤使表面平整化并且去除不在氧化物沟槽中的铜，形成图 4.22(c)所示的结构。



(a) 氧化层图案　　(b) 铜淀积

(c) 平面化后

图 4.22　采用大马士革镶嵌工艺形成铜图案

已经开发**双大马士革**(Dual-Damascene)镶嵌工艺形成铜的通孔。基本的顺序是一样的，只是采用了两次氧化物刻蚀步骤，形成如图 4.23 所示的一般结构。铜的通孔具有比钨更低的电阻，而且也避免了由标准工艺的铝-钨界面引起的接触电阻。

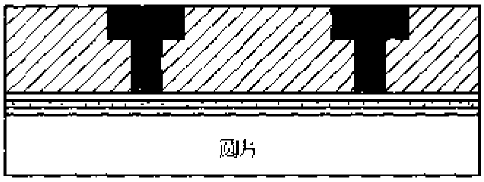


图 4.23 形成铜通孔的双大马士革镶嵌结构

采用铜作为互连材料的另一主要问题是,需要阻止它扩散到硅区域,这是通过运用一层很薄的阻挡层来遏制铜材料而实现的。铜具有较差的粘附特性,所以在铜的周围提供一层阻挡层可以达到这个目的。曾经尝试各种阻挡层材料,它们的结果已发表在文献中,其中包括 W(钨),Ti(钛),TiN,Ta,TaN 和 TaNx(这里 x 为氮的克分子数)。材料的选择会影响互连线的电阻率和薄层电阻,所以需要进行多方面的综合考虑。可靠性问题以及长期效应仍然需要比较仔细地研究。

采用铜引起的制造问题,表明现代硅工艺线的复杂性。甚至一个很小的改变也是有影响的,而一个大改变可能需要好几年的研究才能把它用到生产线上。离子注入曾是一项研究技术,后来才被试验并且重新调整以适合生产线的需要。尽管在时间和金钱上的投资是很大的,但这些例子表明回报也是巨大的。

4.5 设计规则

物理设计的作用是确定一组掩模来定义集成电路。版图本身是运用图形 CAD 工具来完成的,即在计算机屏幕上画出每层上的每个多边形。各层间采用不同的颜色和/或填充图案来区分。图形画在一个参考的栅格上,在每个栅格点之间的距离代表一个规定的长度。设计一个硅芯片的图形非常类似于用一组彩笔在一张格纸上画许多方框。然而,可以画一个东西却并不意味着它一定可以被制造出来。在 IC 制造工艺中采用的每种制造设备都有有限的精度。一台用来成像线宽为 0.25 μm 的光刻机将不能用于 0.18 μm 的成像。对于刻蚀系统也是如此。在硅层次上的物理局限也限制了在硅电路的微电子领域中可以制造什么东西。

拓扑设计规则(DR)是指导版图掩模设计的对几何尺寸的一组规定。一组设计规则规定了最小尺寸、线间距离以及其他几何量的数值,它们是根据各自工艺线的局限制定的。必须遵守这些设计规则,以保证在制造的芯片上有功能正确的结构。图 4.24 所示为一条设计规则的例子,它说明两条靠近的多晶硅线。该图用来说明两个参数:

w_p = 多晶线的最小宽度

s_{p-p} = 多晶线与多晶线间的最小间距

这两个量在设计规则中列出它们的数值;违反这两个数值可能导致设计失败。在工艺中对每一层都赋予一些类似的量。这里的表示法是:

w = 对最小宽度的规定

s = 最小间距值

d = 其他一般的最小距离

并用下标来表示相关的层,例如

w_{m1} = 金属线 1(metal 1)的最小宽度

S_{m1-m1} = 在两条金属 metal 1 线之间的最小间距

这一惯例比较容易理解在版图设计中所规定的和所采用的每条规则。在实际中,每层都编号,而设计规则则通常给予一个与该层编号相应的标识符。

所有的设计规则说明例如 w 和 s 都具有长度单位,而 μm(微米)则是最常用的。例如一种工艺可以规定最小宽度值及间距值为

$$w_p = 0.25\ \mu m,\ s_{p\text{-}p} = 0.425\ \mu m$$

来说明多晶硅的特征尺寸。在 CAD 系统中的版图栅格通常被标定,以保证必要的分辨率。这些值是通过仔细分析生产线的有关部分得到的,且随工艺不同而不同。一组拓扑设计规则可能需要 100 页或更多页的文件来说明,因此需要相当长的时间去了解。这些细节对于制造具有最高可能集成密度的芯片是必不可少的。

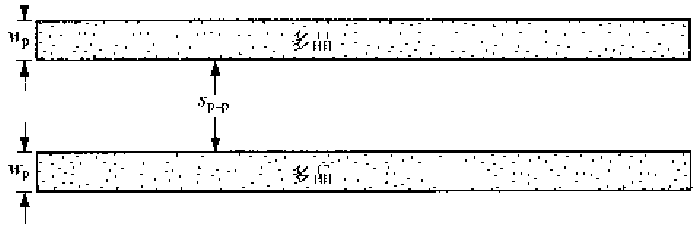


图 4.24 两条多晶线的设计规则

设计规则随工艺进步而改变,因此我们在本书中不介绍具体的一组设计规则。在美国,由政府支持的 MOSIS 集团为大学和小公司提供生产线服务。[③] 读者可以通过**www.mosis.org**进入 MOSIS 网站,查看最新的数据并且下载下来直接使用。考虑到这一点,我们的讨论从本质上讲将保持一般化。

由于现代 VLSI 应用十分普遍,从而产生**硅工厂**的概念。一个硅工厂提供基于付费使用的芯片制造工艺服务。就更广的范围而言,一些工厂如 TSMC 为许多大的公司直到资金充足的个体服务以实现它们的设计。[④] 一个工厂允许设计者提交采用现代工艺的设计。由于用户面很广而且各不相同,大多数工厂运营允许提交一组比较简单设计规则的设计。这些规则可以很容易地缩放,以适应不同的工艺。这样的规则称为 **λ 设计规则**。

λ 设计规则依据一个参照量 λ,它的单位为微米。所有的宽度、间距和距离都写成如下形式:

$$值 = m\lambda \tag{4.21}$$

式中 m 是比例因子。例如规定某层上的最小宽度和间距为 $w = 2\lambda$ 和 $s = 3\lambda$。w 和 s 的数值并不明确,直到 λ 本身被说明为止。如果 $\lambda = 0.15\ \mu m$,那么对这一设计规定的数值就是

$$w = 2(0.15) = 0.30\ \mu m$$
$$s = 3(0.15) = 0.45\ \mu m$$

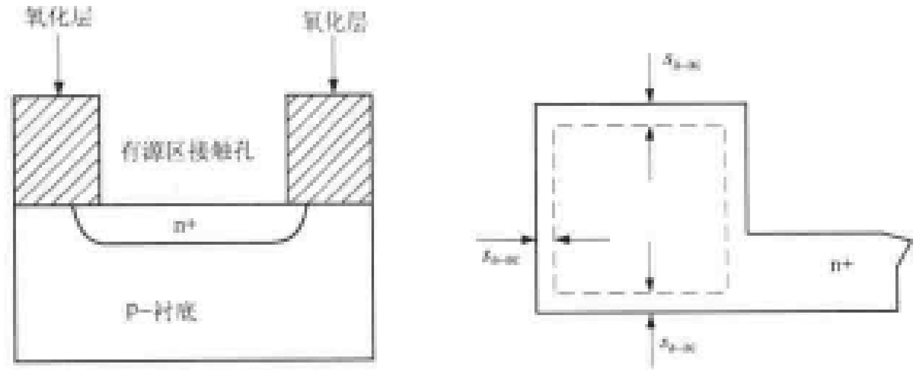
③ MOSIS 代表 MOS Implementation Service。

④ TSMC 代表台湾半导体制造公司。

如果版图设计基于 λ 栅格，那么把设计提交给不同的工艺只需要改变 λ 的数值，而相对尺寸仍维持不变。采用这类**可伸缩**设计规则的主要缺点是运用整数值 m 不可能达到最高的集成密度。

设计规则可以划分成 4 种主要类别：最小宽度，最小间距，离周边距离以及露头。我们已经举了最小宽度和最小间距的例子。当一个特征图形必须放置在芯片表面上已经存在的特征图形的内部时就必须遵守离周边距离这一规则。露头的规则也有类似之处，即它要求图形的一部分必须延伸到一个已有的边界沿之外。

让我们考虑设计一个有源区接触，作为离周边距离规则的例子。如图 4.25(a)所示，氧化层接触刻孔必须对准以使它能处在已经存在的有源 n+ 区域之上。相应的设计规则显示在图 4.25(b)中。在有源区(n+)和有源区接触边缘间的周边距离 s_{a-ac} 必须符合要求，以保证在光刻曝光阶段不会出现没有对准的接触刻孔图案。

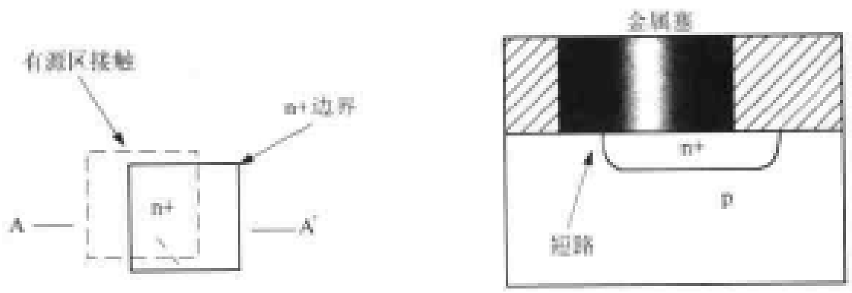


(a) 侧视图　　(b) 离周边距离的设计规则

图 4.25　离周边距离设计规则的例子

在设计规则中必须考虑没有对准好的问题，因为不可能以无限高的精度把掩模图案投射到芯片表面上。套准标记是工艺过程中在某些层上的几何对准图案。这些标记用来对准连续进行的几个形成图案的工序。当淀积一层不透光的材料层时，必须引入一组新的标记。之所以包含离周边距离的设计规则是为了补偿光刻机的对准公差。

图 4.26 表示有源区接触可能出现的问题。假设接触刻孔没有对准落在 n+ 有源区之内如图 4.26(a)所示。在形成接触孔并加入金属塞之后，从图 4.26(b)的横截面图可以看到存在着金属 - 衬底之间的短路。这将使芯片不能正确工作。



(a) 顶视图　　(b) 沿 A–A' 剖面的侧视图

图 4.26　没有对准引起的缺陷

露头类型的设计规则也往往出自于不能对准的问题。作为一个例子考虑形成一个自对准的 nFET。多晶栅用来作为 n 型离子注入的掺杂剂的掩模，它确定漏区和源区。在图 4.27(a)中，包括露头的距离 d_{po}(多晶突出)以保证功能正确的 FET 结构。若不提供突出的距离，那么没有对准的多晶掩模就会导致图 4.27(b)所示的情形。在这一情形下，多晶边沿没有越过整个有源区，所以离子注入后形成了漏—源两边短路。

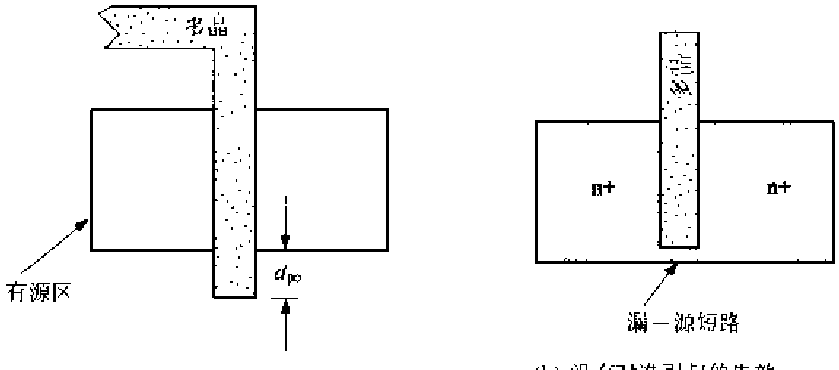


图 4.27 露头(栅突出)设计规则的例子

4.5.1 物理极限

某些几何设计规则是出于实际的考虑，在设计规则中的这些考虑可以是明显的或不明显的。

一个重要方面是成像系统的线宽限制。投射到光刻胶表面上的掩模影像会由于光的衍射而不具有锐边。作为一个简单的估计，具有波长 λ 的光波不能精确地成像比 λ 值小得多的特征尺寸。之所以采用对紫外光(UV)灵敏的正光刻胶是由于紫外光的短波长允许细线宽有较好的分辨率，而且正胶比起负胶来有更好的显影特性。此外，掩模的结构比起已经提到的要复杂得多；一些先进的光学技术，如相移结构，可用来提高分辨率。

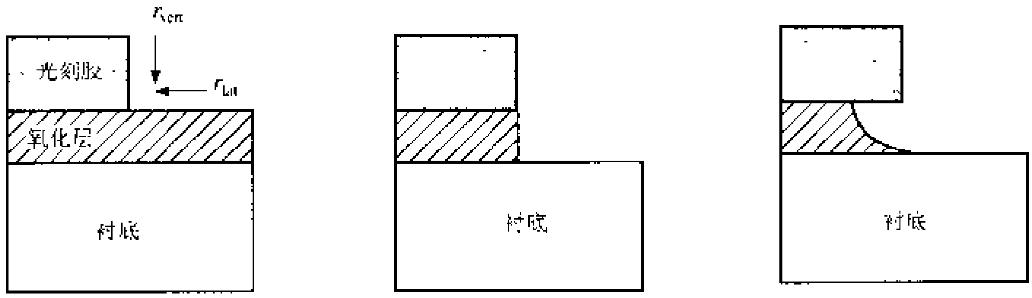
刻蚀工艺会引起另一类问题。当在光刻胶的边沿周围移去材料时，既会发生纵向(垂直于圆片表面)的刻蚀又会发生横向(平行于表面)的刻蚀。可用 $r_{vert}[\mu m/min]$和 $r_{lat}[\mu m/min]$分别表示这两个相应的刻蚀速率，并且定义**各向异性程度** A 如下：

$$A = 1 - \frac{r_{lat}}{r_{vert}} \tag{4.22}$$

由于在 r_{vert}中存在横向刻蚀，这就限制了能够达到的分辨率。图 4.28(a)表示一层氧化层打算通过它上面的光刻胶形成图案。一个纯各向异性的刻蚀剖面显示在图 4.28(b)中。其特征是 $r_{lat}=0$，因而得到垂直壁及 $A=1$。一个 $r_{lat}=r_{vert}$的纯各向同性刻蚀显示在图 4.28(c)中。由于横向刻蚀在光刻胶下部的氧化物被去除从而降低在设计中可以运用的分辨率。引起问题的另一个因素是光刻胶本身的光吸收形态：它使光刻胶的边沿具有一定的斜边，而不是定义得很好的垂直边形状。

硅中半导体的影响也影响设计规则的制定。无论何时形成 pn 结时，它就会在界面上引起所谓的**耗尽区**。根据定义，耗尽区中自由电子和空穴已被“耗尽”，这是由于掺杂剂引起电场并

驱使电荷远离。如果相邻 pn 结的耗尽区相接触，那么阻止电流的特性就会改变，电流有可能在这两者之间流动。这就限定了间隔的规则 s_{n-n}，如图 4.29 所示。该图也显示了必须考虑横向掺杂和各向同性刻蚀影响而应满足的最小线宽参数 w_n。



(a) 光刻胶图案　(b) 纯各向异性刻蚀　(c) 各向同性刻蚀

图 4.28 刻蚀剖面

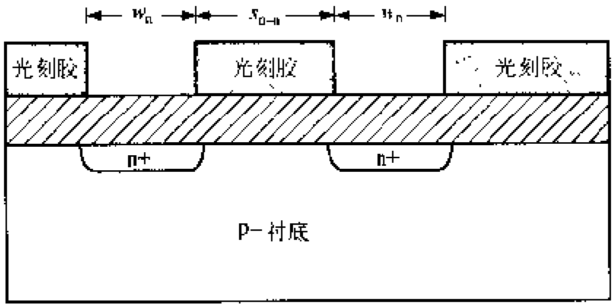


图 4.29 对 n+ 间距的限制

另一个实际的问题是发生在靠得很近的导线之间的电容电气耦合，这会引起所谓的串扰问题，即一部分电能量从一条导线耦合到另一条，引起不希望的扰动称为“噪声”。串扰引起出错，因而是高密度设计中的主要问题。存储器芯片对于这类感应噪声特别敏感。串扰考虑可使设计规则的间隔距离值比光刻所能达到的最小值大得多。串扰问题本身将在第 14 章比较详细地介绍。

4.5.2 电气规则

除了拓扑设计规则外，CMOS 工艺还规定了电气方面的版图规则。这些规则多半是当某些电气状况发生时对基本设计规则的值所做的改变。电气规则可以直接在一般的规则中提供，或作为附录。

电气规则的一个例子是金属互连线的允许宽度。为避免电迁移影响，设计规则将规定给定的线宽允许的最大电流值。较大的电流要求较宽的线。

4.6 参考资料

[1] Stephen A. Campbell, **The Science and Engineering of Microelectronic Fabrication**, Oxford University Press, New York, 1996.

[2] C.Y. Chang and S.M. Sze, **ULSI Technology**, McGraw-Hill, New York, 1996.

[3] James D. Plummer, Michael Deal, and Peter B. Griffin, **Silicon VLSI Technology**, Prentice Hall, Upper Saddle River, NJ, 2000.

第 5 章　物理设计的基本要素

前一章考察了制造 CMOS 集成电路的基本生产工序。本章中将研究把一个逻辑电路转换成硅片的细节，即所谓的“**物理设计**”。一些细节，如一个图案区域所允许的最小尺寸规定这时变得非常关键。然而在 VLSI 芯片物理设计中最重要的学习内容，就是如何使用 CAD 工具以及描述硅掩模的数据库结构。它们提供生产一个芯片所需要的信息，并且提供层次化设计大规模复杂逻辑电路的基础。

5.1　基本概念

物理设计是在硅上产生电路的实际过程。在 VLSI 设计过程的这个阶段，线路图被仔细地转换成几组几何图形，它们用来定义芯片上的物理结构。CMOS 生产工序中的每一层是由不同的图案来定义的。一层图案含有一组几何图形，它们一般称为**多边形**。这自然包括长方形和正方形，但也可以包括有具体尺寸随意复杂的 n 个顶点的多边形。图 5.1 为在 CMOS 设计中见到的这类多边形的例子，它们叠加起来形成整个版图。当它们堆叠成三维结构时，这些层在电气上就等同于电路图。

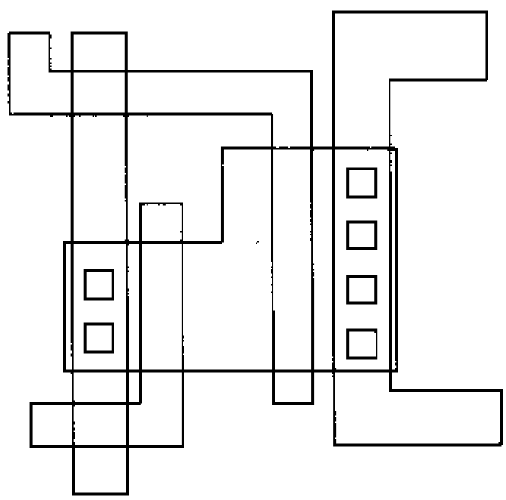


图 5.1　物理设计中多边形的例子

学习到这一点我们已经明白晶体管电路的拓扑连接实现了逻辑功能。换言之，FET 如何用导线连接在一起（串联，并联等）的细节足以决定电路的二进制操作。逻辑的另一个特点是开关速度。它的分析比较复杂，但它对于现代芯片的设计极为重要。尽管这些细节将在以后讨论，但可以在这里总结几个要点。对于给定的一组工艺参数，将发现一个逻辑门的电气特性取决于管子的宽长比。这是由于电气特性同时取决于电流的大小以及器件的寄生电阻和电容。物理设计必须同时考虑这些方面。正如在第 3 章中讨论的那样，必须建立图案以正确实现信号流电路。复杂的因素是每个特征图形的尺寸都会影响电路的电气性能。在一个 VLSI 芯片中，某些门的开关速度将是十分关键的，特别是在长而复杂的逻辑路径中的那些门。在本章中将集中研究电路版图的基础，而在本书第二部分中去探究高速电路中的一些复杂情形。

物理设计的过程是运用称为**版图编辑器**的计算机工具来完成的。这是一个绘图程序，设计者可以用它来说明芯片中每一层上每个多边形的形状、尺寸和位置。为了解决复杂问题，首先是设计简单的门，且把它们的描述存放在一个库的子目录或文件夹中；这些预先设计的门构

成库**单元**。用库单元来构建逻辑块，即通过复制基本单元来构建较大、较复杂的电路。这一过程称为单元**例举**(instancing)，而复制的单元称为**例图**(instance)。

版图必须是逻辑电路的精确表达，但设计者工作的很大一部分是希望以最小的面积得到一个较快的电路。一个多边形形状或面积上的小改变会影响电路的最终电气特性。然而这些改变也许会或也许不会对逻辑链有明显的影响。一个有经验的版图设计者具有一定程度的直观能力，这常常有助于找到故障点。电路模拟也有助于保证版图的正确，从而得到一个符合要求的电路。

CAD 工具

物理设计基于 CAD 工具，它们能够简化设计步骤，帮助验证过程。功能最强的工具集合许多程序，它们联合起来集成一整套设计环境。已有好几套软件系列，每套都有它自己的长处。

让我们通过列出基本芯片设计工具的一些特点，考察它是由什么构成的。对于物理设计过程，最基本的工具是上面已提到的版图编辑器。这是一个联系数据库的图形界面，允许用户画由多边形组成的晶体管和连线。在显示屏上，每层都有不同的颜色或填充图样。每层上的方块或多边形的重叠，成为我们所见的晶体管。版图编辑器建立了每层的数据库，它描述在整个栅格上的图形。这些最终用来在生产过程中制造各层图形所需要的掩模。

在完成版图后，必须运用数据库的信息运行几个辅助程序，确定我们的版图是否正确。设计的电气模拟首先运用一个提取程序，把多边形图形和各层次转换为一个等效的电路。提取程序的输出是一个可以用于电路模拟程序的网表文件；SPICE 是最普遍的网表格式。提取程序得到重要的几何参数，如每个 FET 的设计沟道宽度和长度。它们也说明了晶体管是如何用导线连在一起的。与工艺有关的电气参数也加入提取的输出文件中，形成一个完整的模拟基础。电路模拟程序如 SPICE 通常包括在工具包中(或在一个相关的子目录中)以易于调用。这使设计者可以根据需要立即进行模拟。

一个通常包括在设计环境中相关的程序是"**版图与电路图对照**"(layout versus schematic)检查或者简称为 **LVS**。正如其名，这一程序将版图对照电路图进行检查。验证版图是否相应于所希望的电路是很重要的。LVS 的运行可以运用逻辑图，也可运用电子电路图。

设计规则检查(design rule checker)，简称 **DRC**，是一个运用版图数据库检查在版图上涉及的每条设计规则的程序。这就是说，例如检查在版图中每条金属线的宽度和间距以保证它们不违反所规定的最小值。通过 DRC 保证该设计在生产工艺的限度范围内，可被制造出来。

还有其他工具来帮助完成较大的设计。**布局布线**(place and route)程序帮助版图设计者自动发现在两个给定点之间的可行的布线路径。这在需要把两个复杂单元连接在一起的时候是十分有用的。**电气规则检查**(electrical rule checker)，简称 **ERC**，可以检查电气的连续性，它使连接路径在显示屏上变亮。

这个芯片设计环境的简述为我们提供一个起点，以便比较详细地讨论 VLSI 硅片电路的版图和设计。我们的方法是强调基本的概念和步骤而不去讨论运用任何具体 CAD 工具的细节。一旦理解和掌握了这些技术，它们可以应用到任何环境中去。

5.2　基本结构的版图

让我们从定义芯片中各个区域的顺序开始。讨论将基于在第 4 章中介绍的 p 衬底(n 阱)工艺。[①] 掩模工序为：

0. 从 p 型衬底开始
1. n 阱(nWell)
2. 有源区(Active)
3. 多晶 (Poly)
4. p 选择 (pSelect)
5. n 选择 (nSelect)
6. 有源区接触 (Active contact)
7. 多晶接触 (Poly contact)
8. 金属 1 (Metal 1)
9. 通孔 (Via)
10. 金属 2 (Metal 2)
11. 覆盖玻璃 (Overglass)

应当记住氧化物是在衬底之上在导电层之间生长或淀积的。画芯片版图所需要的细节随顺序而不同。然而只需要很小的修改就可以把这里的概念延伸到任意的工艺线上。

本节将研究如何运用基本的掩模顺序来设计芯片上的基本结构，如 n + 与 p + 区域及 MOSFET。对每个结构介绍相关的设计规则。值得记住的是在每一层上的图形都有设计规则，说明一条线的最小宽度 w 以及在相邻多边形之间的边至边的最小间距。这些表示在图 5.2 中，w 和 s 的实际值取决于在什么层。设计规则只是针对那一层掩模上的图形(掩模是光栅的统称)。芯片上实际制造出来的结构将具有不同的尺寸。

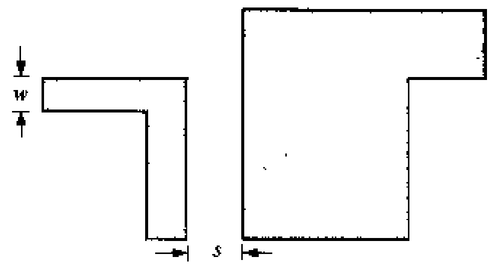


图 5.2　最小线宽和间距

由于这个理由常把版图尺寸称为“**设计尺寸**”，在芯片上最终生产出来的尺寸称为**有效的**或**最终**尺寸。在设计 FET 时这一点特别重要。

我们的讨论将只考虑**曼哈顿**几何形状，即所有的转角都是 90°的倍数。直角的版图是最容易学习的，但它并不总能提供最佳密度。许多版图编辑器允许以随意的方式选择角度，但必须确认这些结构是否为制造过程所支持。

正如在前一章中提到的，讨论在本质上是一般性的。从 MOSIS 网站**www.mosis.org** 可以得到各种工艺设计规则的最新细节。

5.2.1　n 阱

在要制造 pFET 的每个地方都需要 n 阱。我们运用 n 阱掩模来定义 n 阱，在 n 阱掩模上

① 这些工艺是相当一般化的，并且只做很小的改变就可以很容易地扩展到其他工艺。

封闭的多边形代表阱的位置。图 5.3(a)表示两个相邻 n 阱区的横截面。图 5.3(b)中的多边形构成了芯片这一部分的一组掩模。图中表现了两条设计规则：

w_{nw} = n 阱掩模图形的最小宽度

s_{nw-nw} = 相邻 n 阱的边至边的最小间距

常常可把相邻的 n 阱合并在一起成为一个阱。同时必须记住，当一个 n 阱用来制造 pFET 时它必须连接到电源 V_{DD}。

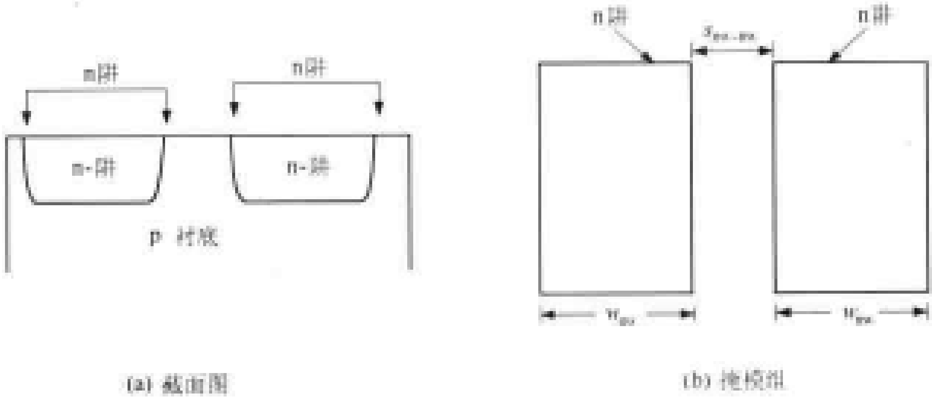


图 5.3 n 阱的结构和掩模

5.2.2 有源区

硅器件是建立在衬底的有源区上的。图 5.4(a)表示一个有源区的横截面。在隔离(场)氧化物生长之后，有源区是平坦的部分并且通向硅圆片顶部。而场氧(FOX)则存在于圆片其余所有的地方。有源区是由**有源区掩模**(Active)上的封闭多边形来定义的。定义图 5.4(a)中的图形所需要的一组多边形表示在图 5.4(b)中。相关的设计规则规定的间距为：

w_a = 一个有源区图形的最小宽度

s_{a-a} = 有源区掩模多边形边至边的最小间距

这些是在最大密度设计中应当保证的最小值。场氧区可以从有源区掩模中由下式推导出来

$$\text{FOX} = \text{NOT (Active)} \tag{5.1}$$

这是一个符号表达式，它基于以下观察：

$$\text{FOX} + \text{Active} = \text{Surface} \tag{5.2}$$

换言之，如果一个区域不是有源区，那么按设定它就是场氧(FOX)。

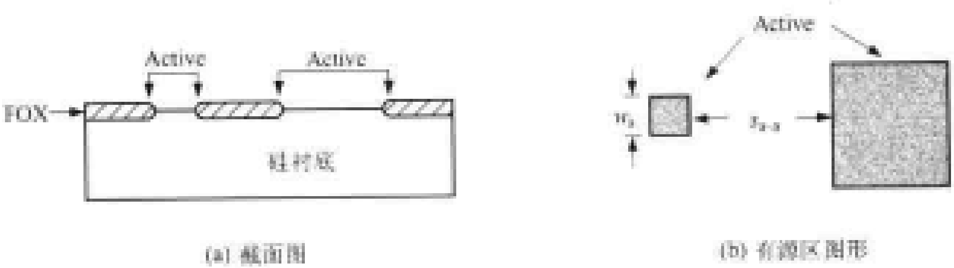


图 5.4 有源区的确定

5.2.3 掺杂硅区

下一步让我们建立 n+ 和 p+ 区。这些也分别称为 **ndiff** 和 **pdiff**，它们的名字来源于在过去时候杂质是运用一种称为**扩散**的热学技术而不是用离子注入引入到圆片中的。图 5.5(a) 表示了一个 n+ 区。它的形成是用离子注入把砷或磷离子注入到由 **nSelect** 掩模规定的衬底区域中。由于这一步是在隔离工艺之后完成的，所以 nSelect 掩模定义了覆盖包含有源区的区域。图 5.5(b)的一组掩模同时表示了为构造 n+ 区所需要的 nSelect 和有源区。让我们用表达式(Mask-name)来表示在该层上所有的一组多边形。如果只包含 nSelect 和 Active 掩模，可把 n+ 区域表示成：②

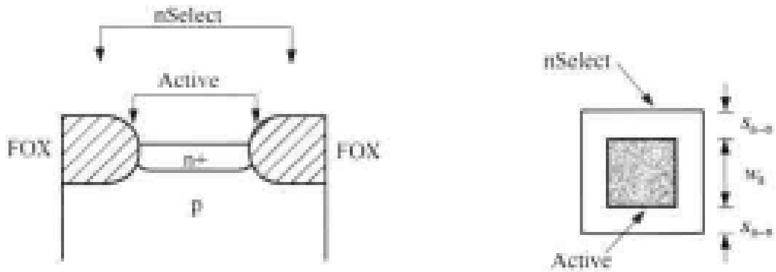
$$\mathrm{n+} = (\,\mathrm{nSelect}\,) \cap (\,\mathrm{Active}\,) \tag{5.3}$$

这就是说，无论何时当有源区掩模和 nSelect 掩模交叉时就产生 n+ 区域，这是用交集运算符 ∩ 来表示的。图中表现了两条设计规则，它们是

w_a = 一个有源区的最小宽度

s_{a-n} = 有源区至 nSelect 间的最小间距

式中间隔距离是从边到边；这一惯例以后将在全书中应用。设计规则通常对不同的方向是不变的，所以同样的值也适用于水平的尺寸。



(a) 截面图　　(b) 掩模组

图 5.5　n+ 区的设计

p+ 区域通过离子注入把硼注入圆片上有源区的开口处得到。图 5.6 (a)的截面图表示用这一技术在 n 阱区形成 p+ 区。形成 p 型有源区是由 **pSelect** 掩模定义的离子注入实现的。图 5.6(b)表示所要求的一组掩模，图中为了完整起见也画出了 **nWell** 掩模。P+ 区的表达式为：

$$\mathrm{p+} = (\,\mathrm{pSelect}\,) \cap (\,\mathrm{Active}\,) \cap (\,\mathrm{nWell}\,) \tag{5.4}$$

这里只考虑这三种掩模。这就是说，无论何时当 pSelect 掩模和 Active 掩模区在 nWell 区内重叠时就形成了 p+ 区。设计规则规定的重要间距为

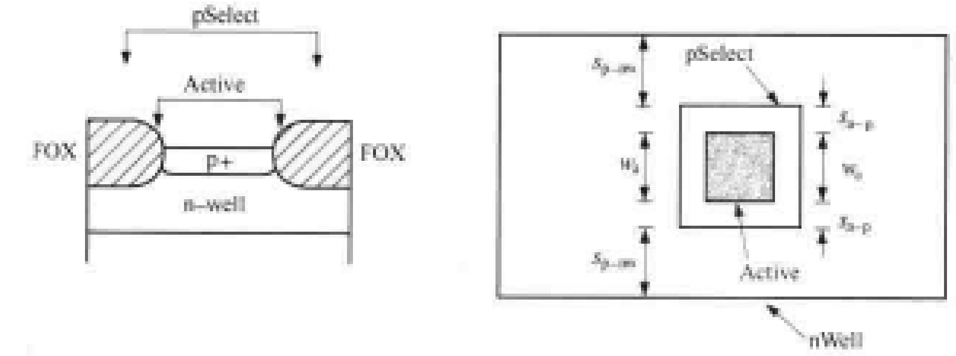
w_a = 有源区最小宽度

s_{a-p} = 有源区至 pSelect 的最小间距

② 在 Magic 版图编辑器中，ndiff 和 pdiff 是用一个命令画的，所以单独的 Active 和 nSelect 图案是没有必要的。然而，注意在工艺中却没有像 ndiff 和 pdiff 这样的掩模。

$$s_{p-nw} = \text{pSelect 至 nWell 的最小间距}$$

同样,这些也是由一组工艺设计规则来规定的



(a) 截面图　　(b) 掩模组

图 5.6　p+区的设计

5.2.4　MOSFET

每当一条多晶栅线完全越过 n+或 p+区域时,就会形成自对准的 MOSFET 结构。实际上,多晶线是在离子注入之前淀积的,而且它的作用可以阻止掺杂剂进入硅中。因此 FET 要求运用在**多晶掩模层**上的多边形。对多晶图形的基本设计规则是

$$w_p = \text{多晶最小宽度}$$

$$s_{p-p} = \text{多晶至多晶的最小间距}$$

最小的多晶线宽 w_p 与 FET 的设计沟道长度相同。

让我们首先构建一个 nFET。图 5.7 (a)的横截面图表示 n+ 和多晶层;在栅和衬底之间的栅氧没有明显地显示。图 5.7 (b)的顶视图显示了管子的设计沟长 L 和沟宽 W 值。为了建立一组掩模,只是把一个多边形加到多晶掩模上,把 n+ 区分隔成两个区域。于是就形成了图 5.8 所示的掩模。这意味着设计规则为:

$$L = w_p = \text{多晶线的最小宽度}$$

而所示的另一个设计规则为:

$$d_{po} = \text{多晶离开有源区的最短露头}$$

这条规则要求保证当在光刻过程中发生小的套准误差时仍能形成自对准 FET。这称为**栅的突出距离**。用这一图可以写出 nFET 的中心部分的定义:

$$\text{nFET} = (\text{nSelect}) \cap (\text{Active}) \cap (\text{Poly}) \tag{5.5}$$

因为这就是沟道形成的地方。n+区定义为

$$\text{n+} = (\text{nSelect}) \cap (\text{Active}) \cap (\text{NOT [Poly]}) \tag{5.6}$$

这比起上面在方程(5.3)中给出的定义更为精确,在式(5.3)中忽略了存在多晶掩模。

pFET 也是以同样的方式形成的。图 5.9(a)为这一器件的横截面图,而图 5.9(b)的顶视图则为在掩模上画出的重要的沟道尺寸 L 和 W。值得提及的是 n 阱区域为隐含存在的 p 衬底所包围,这在顶视图中明显地表示出来。图 5.10 中所示的 pFET 一组掩模与 nFET 具有相

同的基本特点，差别只是在离子注入的极性（pSelect 而不是 nSelect）以及在管子周围存在 n 阱。所画的沟道长度 L 相应于最小的多晶线宽，而 d_{po} 是关于栅突出的设计规则。由该图所包括的其他设计规则已经讨论过。pFET 中心区的简单表达式为：

$$pFET = (pSelect) \cap (Active) \cap (Poly) \cap (nWell) \qquad (5.7)$$

这表明这一器件是四层掩模的重叠部分。因此 p+ 区可描写为：

$$p+ = (pSelect) \cap (Active) \cap (nWell) \cap (NOT\ [Poly]) \qquad (5.8)$$

这也就是该器件中没有形成多晶的地方。它比起式(5.4)的较为简单的表达式来更为精确，而式(5.4)忽略了多晶掩模。

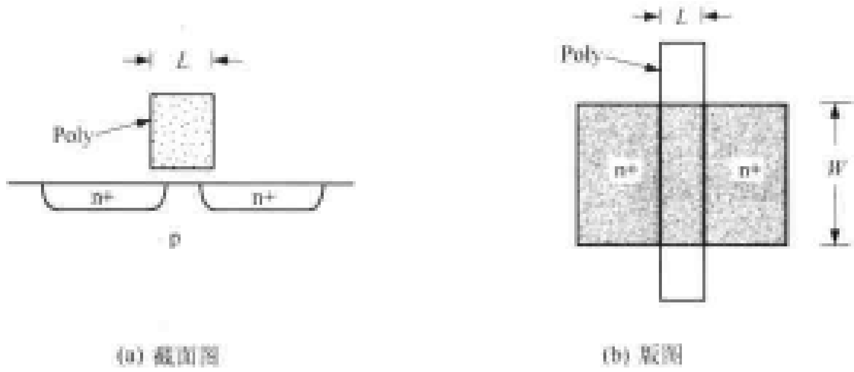


图 5.7 nFET 结构

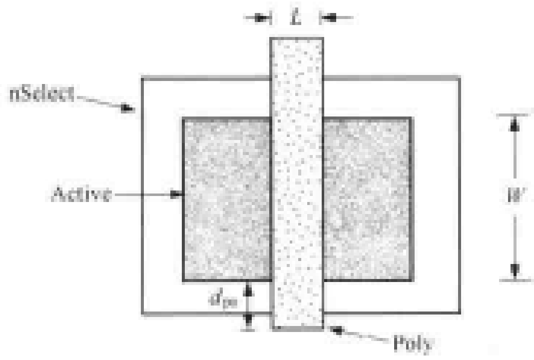


图 5.8 nFET 掩模

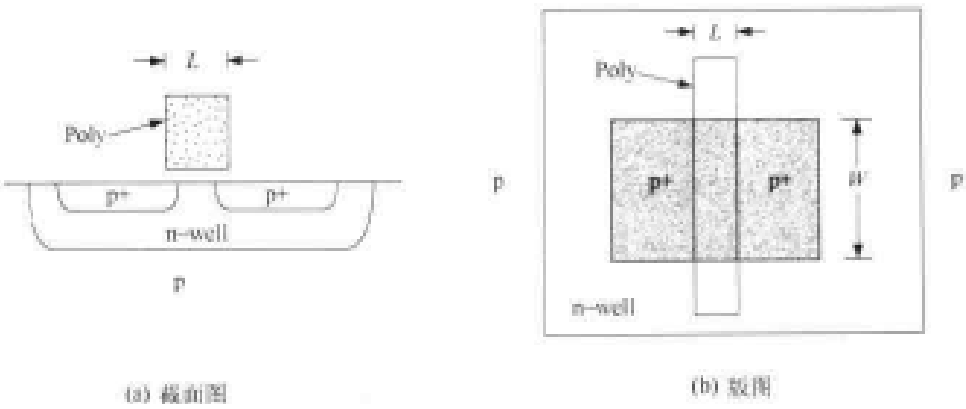


图 5.9 pFET 结构

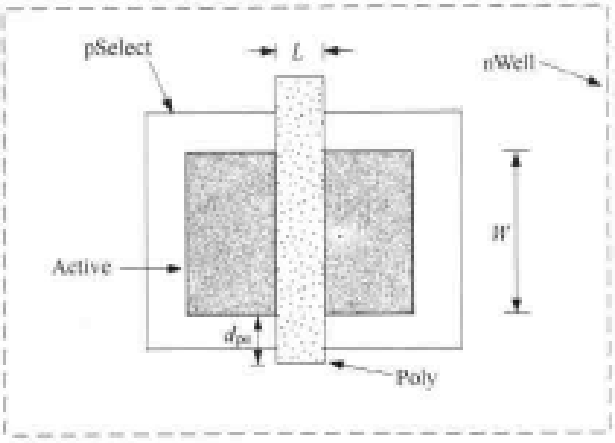


图 5.10 pFET 掩模组

MOSFET 的设计值及有效值

MOSFET 的关键尺寸是沟长 L 和沟宽 W。正如已经看到的，L 是由多晶栅线的宽度确定的。如果追溯一下制造顺序就会发现沟宽 W 是由晶体管有源区的边长确定的，因为这一区域定义了漏/源离子注入穿入硅中的地方。正如在前一章中提到的，设计规则限定了掩模版图的设计，它所规定的尺寸是指设计尺寸，而在芯片上测量到的最终尺寸会有所不同。在对管子进行电气分析时，这两个尺寸间确切的相互关系特别重要。

让我们来考察图 5.11(a)所示的 FET 版图几何形状；这个一般的版图同时适合于 nFET 和 pFET。首先考虑器件的沟长。图中所画的 L 值是多晶硅的线宽。然而最终结构中在两个 n+ 区域间的尺寸要小于 L，这是由于在注入退火步骤期间的横向掺杂所致。当圆片被加热时，在源漏两边的掺杂剂各自朝着另一边运动。重叠的效应是对称的，所以在两边形成了相同的重叠距离 L_o。在对管子进行电气分析时，重要的距离是在两个 n+ 区域之间的最终实际值。若需要加以区分时，这个最终实际值称为**电气或有效沟长**。若将有效沟长表示成 L_{eff}，则可看到：

$$L_{eff} = L - 2L_o \tag{5.9}$$

上式给出了它的数值。一个更为一般的形式是

$$L_{eff} = L - \Delta L \tag{5.10}$$

式中 ΔL 为由于重叠和其他影响引起的沟道长度的总的减少。

由于生长场氧引起有源区域减少，沟道宽度也会小于设计值。这称为有源区的**侵蚀**(encroachment)，它使有效的沟宽为

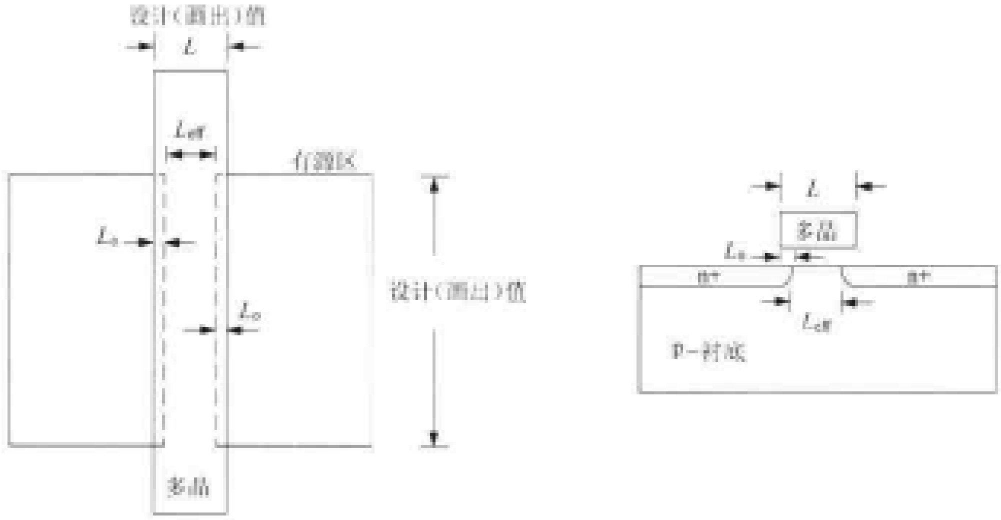
$$W_{eff} = W - \Delta W \tag{5.11}$$

式中 W 是设计值，而 ΔW 是由于所有的影响而引起的沟宽(原文为沟长——译者注)的减小。分析电气特性时管子的宽长比总是用有效值比，即

$$\frac{W_{eff}}{L_{eff}} \tag{5.12}$$

而不是设计尺寸的比(W/L)。当运用公式求某些量的值，例如求 nFET 的电阻 R_n 时，记住这

一点是很重要的。如果采用电路模拟的 CAD 工具，一般采用设计尺寸而让程序去计算有效值。这将在本书中以后谈及 SPICE 时再进行讨论。

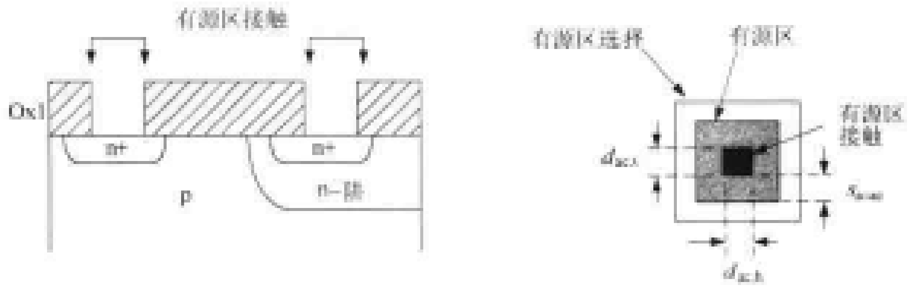


(a) 设计（画出）的版图　　(b) 最终制造出的情形

图 5.11　MOSFET 的设计(画出的)尺寸及有效尺寸

5.2.5　有源区接触

一个有源区接触是在氧化物 Ox1 上刻孔，使第一层金属能够接触 n+ 或 p+ 有源区，它表示在图 5.12 (a)的横截面图中。这些接触是由**有源区接触**掩模及通常的覆盖所定义的，如图 5.12 (b)所示。因为接触是放置在一个 n+ 或 p+ 区域之内，因此它要服从有关周围距离的设计规则：



(a) 截面图　　(b) 一般情况的掩模组

图 5.12　有源区接触的形成

$$s_{a-ac} = \text{有源区和有源区接触之间的最小间距}$$

接触尺寸：

$$d_{ac,v} = \text{接触垂直方向的尺寸}$$
$$d_{ac,h} = \text{接触水平方向的尺寸}$$

是确切的规定。当

$$d_{ac,v} = d_{ac,h} = d_{ac} \tag{5.13}$$

时就会形成一个正方形的接触。但具有不同于 1:1 的宽长比的接触也是很普遍的。

5.2.6 金属层 1

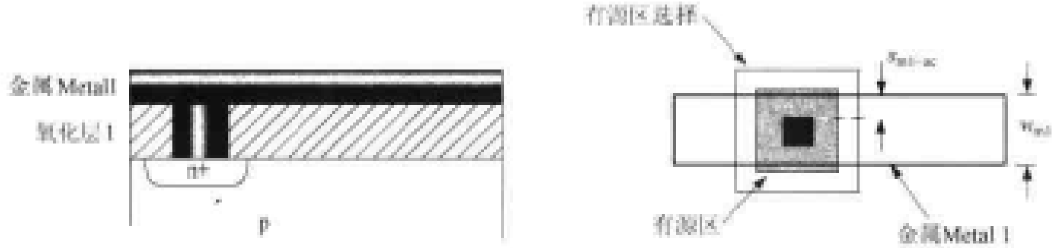
金属层 1(Metal 1)在 Ox1 氧化层之后加到圆片上,它用来作为信号互连线,也用做分配电源线。图 5.13(a)表示第一层金属线及一个连至 n+ 区的有源区接触的横截面图。穿过氧化层的接触刻孔中填充一个塞,如在前一章中描述的那样。图 5.13(b)为这一结构的一组掩模,其中 **Metal 1** 掩模图形与有源区接触重叠以形成电气连接。图中表示了两条设计规则:

$$w_{m1} = \text{Metal 1 线的最小宽度}$$

及

$$s_{m1-ac} = \text{Metal 1 至有源区接触的最小间距}$$

此外,Metal 1 之间还有一个规则规定的最小间距 s_{m1-m1} 值,但它没有明确表示在图中。



(a) 截面图　　(b) 一般情况的掩模组

图 5.13　金属线与有源区接触

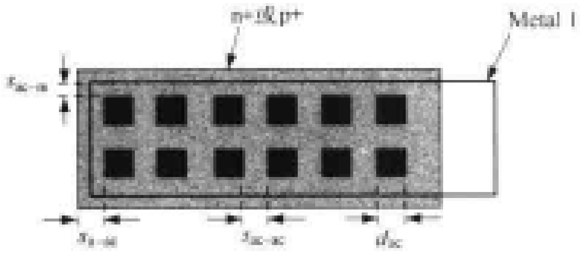


图 5.14　采用多个接触来降低接触电阻

每个接触的特征是它的电阻:

$$R_C = \text{接触电阻}(\Omega)$$

这个电阻是由于金属连接造成的。为了限制总的电阻,通常采用设计规则允许的尽可能多的接触。图 5.14 为一个例子。因为这些接触都是并联在一起的,所以有 N 个接触的 Metal 1 至有源区接触的连接的等效电阻为

$$R_{c,eff} = \frac{1}{N} R_c \tag{5.14}$$

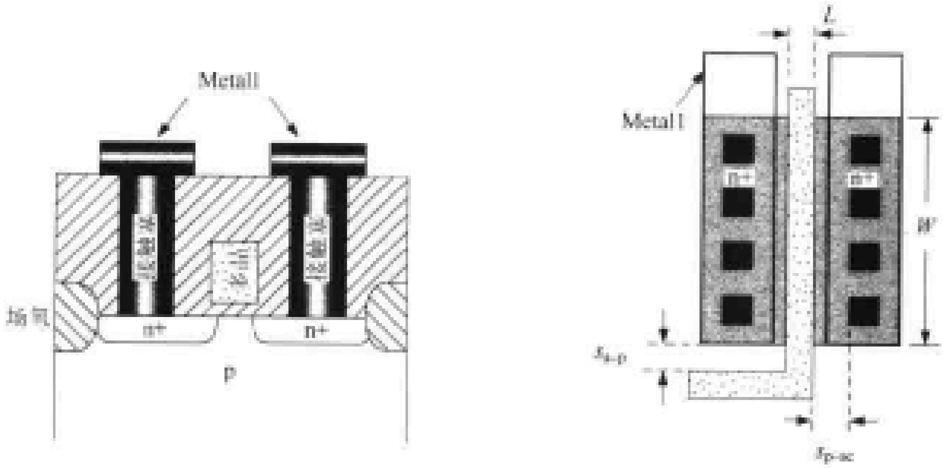
在本例中,$N = 16$,所以这一连接的等效电阻为单个接触电阻值的(1/16)。这些接触也使电流分开流动。

Metal 1 可以通过有源区接触的氧化层刻孔,连接到 MOSFET 的有源区上。漏和源端通常处在 Metal 1 层上,如图 5.15(a)中所示的 nFET。相应的版图显示在图 5.15(b)中,这里我们已包括了以下设计规则:

s_{p-ac} = 从多晶至有源区接触的最小间距

s_{a-p} = 从有源区至多晶的最小间距

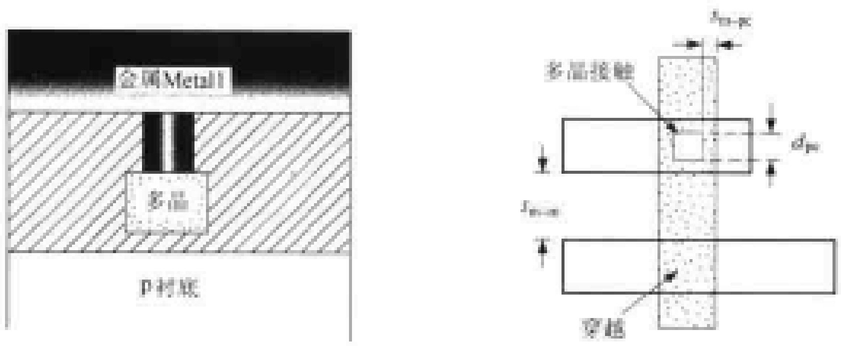
第 1 个参数规定了与周边的距离以确保有源区接触不会破坏任何多晶栅。第 2 个参数规定的间隔距离 s_{a-p} 是考虑自对准 FET 工序；它保证即使多晶掩模没有准确地对准圆片上形成的有源区图案，FET 也有合适的尺寸。



(a) 截面图　　(b) 版图

图 5.15　FET 漏和源端与 Metal 1 的连接

多晶接触掩模可以用来在 Metal 1 和多晶栅之间形成电气连接。图 5.16(a)所示是在这两层之间接触的横截面图。多晶接触掩模定义在图 5.16(b)版图上部显示的中空正方形所表示的氧化层刻孔。在版图的下部，Metal 1 和多晶层并没有连接。这一"穿越"的特点对于布线是非常有用的。



(a) 截面图　　(b) 版图

图 5.16　多晶接触

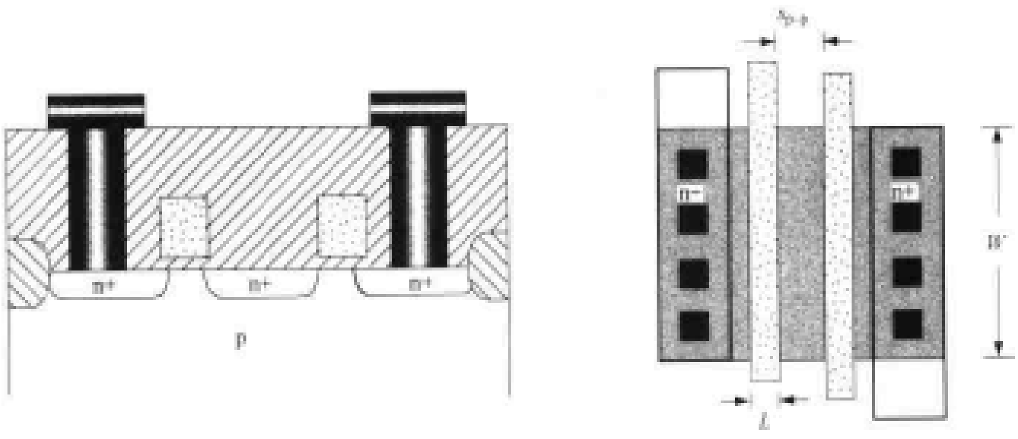
作为最后一个例子，让我们来构建一对串联的 FET。图 5.17(a)所示为两个 nFET 的横截面图。串联连接通过共享中间的 n+ 区实现；因为 n+ 是较好的导体，因此在这两个器件之间不需要另外的连线。图 5.17(b)的版图采用了这一特点：串联晶体管由平行的多晶线形成。重要的设计规则规定的间距为：

s_{p-p} = 多晶至多晶间的最小间距

为了得到一对并联的 FET，增加了接触孔，如图 5.18 所示。图中表示的间距 s_{g-g} 是在两条栅之间的距离。这并不是一条设计规则，但它可用至今介绍的基本设计规则写成如下形式：

$$s_{g\text{-}g} = d_{ac} + 2s_{p\text{-}ac} \tag{5.15}$$

因为必须保证接触本身的尺寸再加上两个多晶至有源区的间距，但这不是应用到每种工艺的一般规则。在有些亚微米的设计中采用多晶与多晶间的距离 s_{p-p} 而不管这种情形；接触可以在两条栅之间加入而不必增加它们间的距离。



(a) 截面图　　(b) 版图

图 5.17　串联的 FET

当采用公共的有源区形成具有不同 W 值的 FET 时，需要引入另一条设计规则。图 5.19 表示两个串联的 nFET，它们的沟道宽度 $W_2 > W_1$。多晶至有源区的间距 s_{p-a} 是指一条栅的边与有源区边界改变处之间的距离。在这一设计中这条规则要应用两次，因为两个 FET 都存在有源区边界改变的情形。

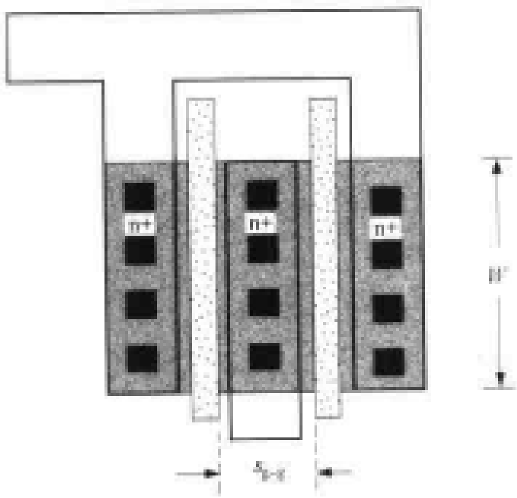


图 5.18　并联的 nFET

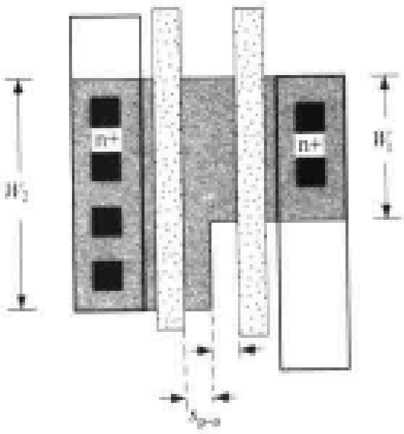


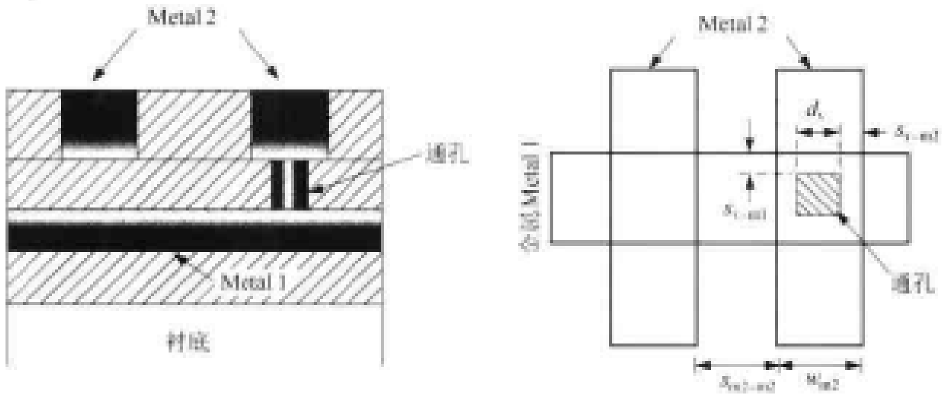
图 5.19　同一有源区形成不同的沟宽

5.2.7 通孔和多层金属

尽管简单的电路可用单层多晶和单层金属工艺形成，但在复杂的电路中互连线的布线变得非常困难。现代的 CMOS 工艺增加了另外的金属层用来分布信号和电源。我们将根据金属层被增加的顺序来标记它们。例如在 4 层金属的工艺中，层次的顺序可以是：

$$\text{Metal 1} \rightarrow \text{Metal 2} \rightarrow \text{Metal 3} \rightarrow \text{Metal 4}$$

CVD 氧化层淀积在金属层之间，使每层在电气上是不同的。在相邻金属层之间的连接采用通孔掩模来实现。这相当于一个有源区接触掩模定义氧化层刻孔的位置；在通孔中填以金属插塞以实现在两层金属间的电气接触。



(a) 截面图　　(b) 版图

图 5.20　用通孔实现 Metal 1 与 Metal 2 之间的连接

图 5.20(a)说明了运用通孔来连接 Metal 1 至 Metal 2。若没有通孔（如该图的左边部分），这两层金属层在电气上是隔离的，图右边的通孔提供了在这两层之间的连接。图 5.20(b)为掩模版图，所表现的新设计规则为：

d_v = 一个通孔的尺寸（在垂直方向上可以不一样）

w_{m2} = Metal 2 图形的最小宽度

s_{m2-m2} = 相邻的 Metal 2 图形间的最小间距

s_{v-m1} = 在通孔和 Metal 1 边之间的最小间距

s_{v-m2} = 在通孔和 Metal 2 边之间的最小间距

在其他金属层之间的通孔也类似。注意到在 $j>1$ 时第 j 层金属层的 w_{mj} 和 s_{mj-mj} 值是不同的，这是因为圆片表面的布局和不平度常常要求采用较宽的线。

5.2.8 防止闩锁现象

闩锁(Latch-up)是可能发生在用体硅 CMOS 工艺生产的电路中的一种情况。当一个芯片处在闩锁状态时，它会从电源吸取很大的电流，但对输入激励却没有响应而不能正确工作。一个芯片可能先是正常工作而后进入闩锁状态；在这种情形中，移去并且重新连接电源有可能恢复工作。在最坏情况下，当接通电源时芯片就可能进入闩锁状态而根本不能正确工作。如果电

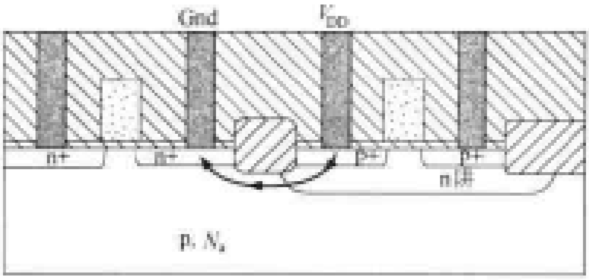


图 5.21 闩锁时的电流路径

流太大，那么热损耗将破坏芯片。

图 5.21 表示当芯片处在闩锁时的电流路径。当条件合适时，这一路径具有非常低的电阻而可能产生很大的电流。理解闩锁的关键是注意体硅工艺在电源 V_{DD}和地之间产生了**4 层的 pnpn** 结构。如图 5.22(a)所示，这一结构具有图 5.22(b)所示的电流－电压关系。对于小的 V_{DD}电压，电流 I_{DD}很小，这是因为 pn 结反向不导通的特点所致。如果 V_{DD}达到一个**转折电压** V_{BO}时，这个反向阻断特性由于内部电场而被破坏。于是，就有如图所示的大电流，这表明芯片已进入闩锁状态。

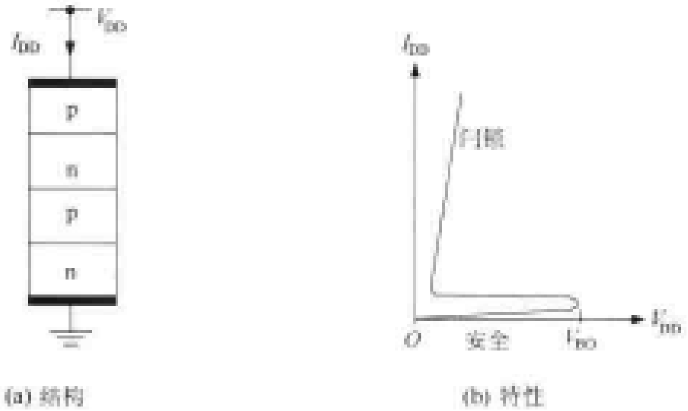


图 5.22 4 层 pnpn 器件的特性

在物理设计层次上就可以设计阻止闩锁，即可以采用各种规则来避免形成电流通路。有个概念非常简单：由于电流必须流过 n 阱和 p 衬底，可以在许多不同的位置上放上 V_{DD}和接地连接，以使电流绕开这个"坏"路径。这就给我们以下一般的规则：

- 每当一个 pFET 连到电源 V_{DD}时则放置一个 n 阱接触
- 每当一个 nFET 连到接地端时则放置一个 p 衬底的接触

因为总是要进行电连接的，所以记住它们是一件简单的事情。这些表示在图 5.23 中，它对避免闩锁非常有效。也已开发了其他技术，设计者应当总去查看设计规则中关于闩锁怎样才能避免的规定。

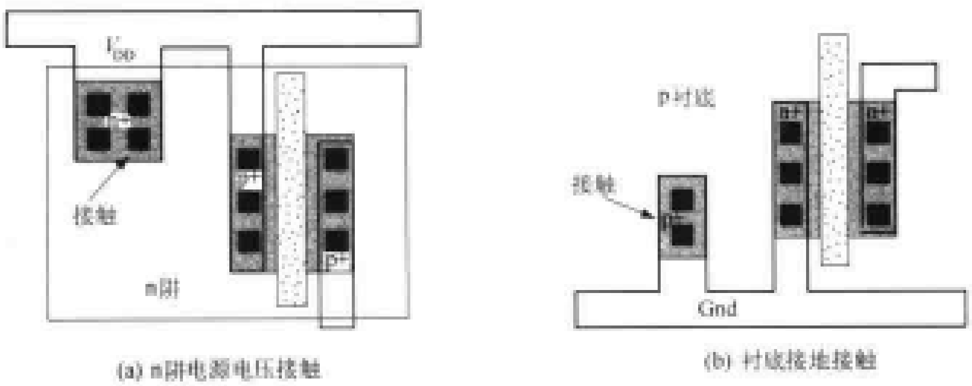


图 5.23 采用 n 阱和衬底接触防止闩锁

不直接在硅衬底上构造管子的非体硅 CMOS 工艺，通过不形成 pnpn 层避免了闩锁问题。绝缘体上硅（SOI）的设计就是如此。此外 FET 采用两个单独的阱，一个 n 阱放 pFET 及一个 p 阱放 nFET，可以帮助阻止电流路径的形成。这些**双阱**工艺在先进的工艺线上是非常普遍的。

由于闩锁是由高电压引起的，因此在设计具有较高感应"噪声"电平的电路，例如数据接收电路时必须特别小心。有关避免这类问题的信息也包含在一组设计规则中。一个新的设计者往往并不担心闩锁，直到一个芯片由于闩锁而失效；从这点看，这个问题是值得引起重视的！

5.2.9 版图编辑器

本节已经介绍了版图的几个重要特点。比较关键的几项总结在下面可以作为参考。

- 无论何时当有源区为 nSelect 包围时就形成 n+；这也称为 ndiff。
- 无论何时当有源区为 pSelect 包围时就形成 p+；这也称为 pdiff。
- 无论何时当多晶把一个 n+ 区域划分成两个分开的部分时就形成 nFET。
- 无论何时当多晶把一个 p+ 区域划分成两个分开的部分时就形成 pFET。
- 在导电层（n+，p+，多晶，金属等）之间不存在电流路径，除非提供一个接触刻孔（有源区接触，多晶接触或通孔）。

这些简单的观察为我们将要遇到的大多数版图问题提供了基础知识。

版图编辑器通过对每层定义不同的颜色和/或填充样式以便在看图时加以区分。这里只采用不同的灰度和不同的线宽以节省彩印本书所需要的成本。[5] 图 5.24 是本书中用来识别各个层次的图例。注意 n+ 和 p+ 区具有相同的图例，所以一个区域的极性是由它所处的位置隐含确定的：若它在 n 阱中则为 p+ 区，否则它就是 n+ 区。

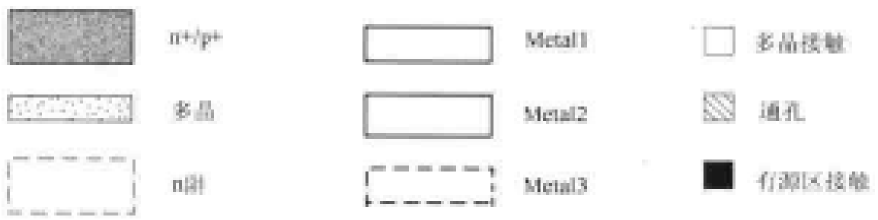


图 5.24　本书中版图的各层图例

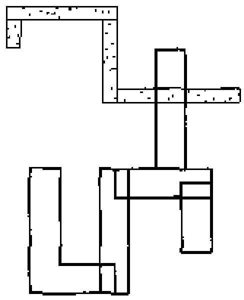
每种版图编辑器的工作略有不同，但它们都有相同的基本特点。

- 画一个多边形时首先选择所希望的材料层，然后运用画图工具按要求画出它的形状。
- 版图编辑器提供一个栅格背景。在每个栅格点之间的距离是一个预先规定的距离。
- 各层可以按任何次序来画，只要每个多边形能按层次的颜色/名称/图案被正确地识别。数据库自动跟踪在每层上所画的多边形。
- 版图图案用来建立为工艺所需要的一组掩模，这些图案也包括了所画尺寸的信息。
- 必须遵守设计规则，在版图完成之前必须检查间距。
- 在一个给定层上的多边形可以画成相接的或重叠的。重要的只是轮廓线，这表示在图

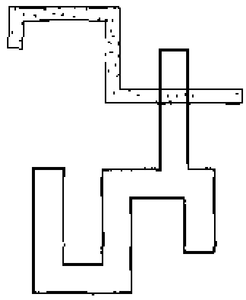
⑤　彩印本是本书价格的 4 倍。

5.25 中。图 5.25(a)的整个版图是用矩形画出的,但最终完成的掩模如图 5.25(b)所示。这简化了整个版图过程。

一定要及时存储你的设计!当芯片(的版图)完成时,它通常以一种标准的格式送到工艺线。为了与最早的芯片设计者说法一致,这一过程称为“tape - out”(**记带**),因为那时版图文件是记在磁带上送到生产线的。最常用的文件格式也许就是 GDS 标准格式,它是早期以微型计算机为基础的 CAD 系统的格式标准。学术界的用户常常生成 CIF 格式(Caltech 中间格式)的文件,这一格式是在 20 世纪 70 年代开发的。



(a) 多个多边形



(b) 掩模图形

图 5.25 用矩形画复杂的多边形

5.3 单元概念

数字 VLSI 芯片是基于层次化设计的概念。用一个个晶体管来构成逻辑门,然后用这些门来构成逻辑链和功能块,又用来作为更大单元的基础。物理设计中基本的建筑块称为“**单元**”。一个单元可以像一个 FET 那么简单,或像一个运算逻辑单元(ALU)那么复杂。不论内部有多复杂,每个单元起作用的方式是相同的:它可作为部件构造更大的逻辑电路。

基于单元设计的主要概念是很容易明白的。假设从一组 CMOS 逻辑门(NOT, NAND2, NOR2)开始,且为它们每一个设计物理版图。在最底层,集中在对每一层按所要求的尺寸放置多边形。“后退”一步把这些门看成如图 5.26 所画的那样,每个功能块是一个独立的单元。在设计层次的这一层上,我们并不关心内部细节,只有门的外部特性才是重要的,这里已把整个版图用一个等效的逻辑符号来替换。在所示的例子中,输入和输出端表示为进入这个单元的**端口**。一个端口用来访问内部的电路。同时注意到一个单元需要 V_{DD}和 V_{SS}的电源端,它们对每个单元都被选择在相同的位置上。最后,每个单元即 NOT, NAND2 及 NOR2 的宽度分别表示为 X_{NOT}, X_{NAND2}及 X_{NOR2}。它们的数值取决于在物理层次上运用的管子尺寸和布线。

一旦定义了一组单元,它们就可用来构造更为复杂的电路。假设希望一个单元提供如下的功能

$$f = \bar{a} \cdot b \tag{5.16}$$

这可以如图 5.27(a)所示那样用两个 NOT 门和一个 NAND2 门简单地串联起来就可实现。

Metal 1 导线按要求把单元的端口连接起来。例如第一个 NOT 门的输出连到 NAND2 门的 in 1。一旦构成了这个串联链，可以定义如图 5.27(b)所示的新单元 F1。这个新单元的总宽度为

$$2X_{NOT} + X_{NAND2} \tag{5.17}$$

它即是构造它的三个单元的宽度之和。一旦定义了这个新单元 F1，它就可以作为一个建筑块来用，而不需要把它分解成最初构建它的几个单元。它就成了像 NOT，NAND2 和 NOR 电路一样的基本单元。采用这种层次化的设计方法，可以设计和构造极为复杂的逻辑电路。事实上，它是在 VLSI 中需要学习的最重要的技术之一。

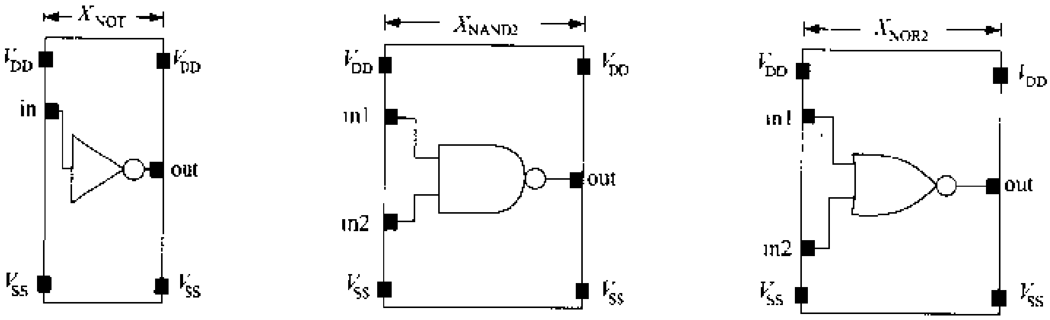
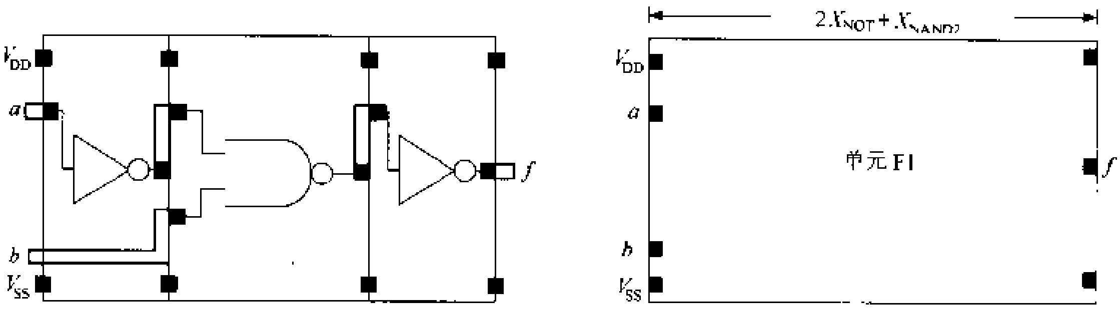


图 5.26　作为基本单元的逻辑门



(a) 基本单元　　(b) 新的复杂单元

图 5.27　用基本单元构成新单元

让我们把注意力集中在物理层上如何构成基本的单元集合。第 1 个应当研究的内容是如何放置电源线 V_{DD}和 V_{SS}。这个问题表示在图 5.28 上。电源和地线都显示在 Metal 1 层上。在这两条线之间的间距为

$$D_{m1-m1} = \text{在 } V_{DD}\text{和 } V_{SS}\text{之间的边到边的距离}$$

而节距(pitch)为：

$$P_{m1-m1} = \text{在 } V_{DD}\text{和 } V_{SS}\text{线的中线之间的距离}$$

这两者的关系是

$$P_{m1\text{-}m1} = D_{m1\text{-}m1} + w_{DD} \tag{5.18}$$

式中 w_{DD}是电源线的宽度。[4] 制造业的专家常常运用节距说明，而边之间的实际距离 D 对于电路版图比较有用。用于 pFET 的 n 阱区放在 V_{DD}线的附近，如图所示。而在 V_{SS}附近的

① 注意 w_{DD}可以大于 Metal 1 线设计规则允许的最小宽度 w_{m1}。

区域留给 p 衬底，因为 nFET 连接到 V_{SS}。

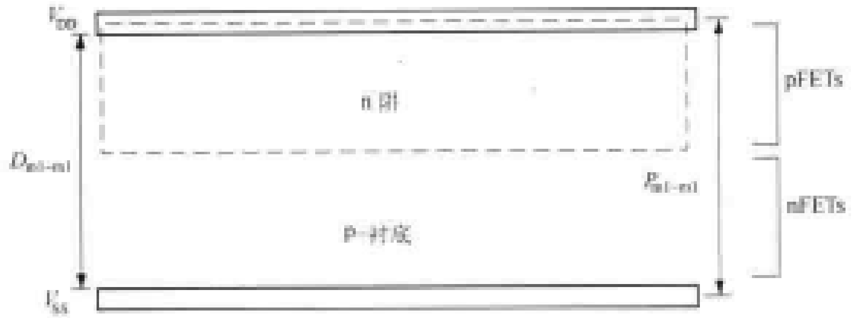


图 5.28 V_{DD}和 V_{SS}电源线

一旦布置了 V_{DD}和 V_{SS}线，可以着手在它们之间放置 FET。图 5.29 显示了用两种不同的方法确定管子的方向。图左边的 FET 的漏源沿水平方向，FET 沟宽 W_n 和 W_p 是由 D_{m1-m1} 和 n 阱尺寸限制的。如果像图右边所示把 FET 旋转 90°至垂直方向，那么沟宽 W_n 和 W_p 可按需要选择任意尺寸，然而单元的宽度会变大。因为我们希望每个单元选择同一组 D_{m1-m1} 值，应当研究 FET 的位置对单元大小的影响。

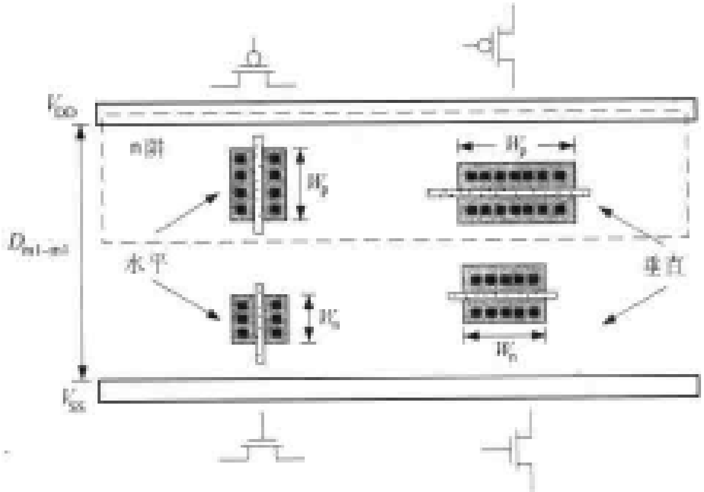


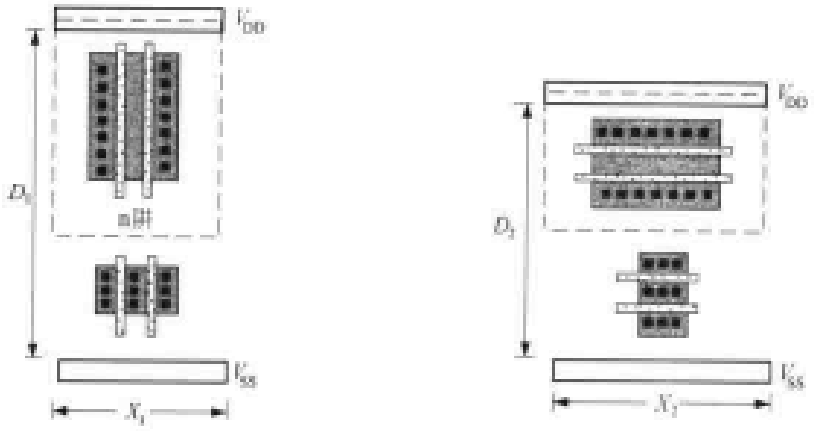
图 5.29 MOSFET 的定向

以上两种 FET 位置各自的优缺点表现在图 5.30 中。图 5.30(a)所示为水平放置的管子，希望能使 D_1 足够大，以便包含所需要的最复杂的门。如果采用垂直的管子，那么图 5.30(b)所示的 D_2 值可以比 D_1 小。差别就在于单元的水平宽度。一般地，对于一个给定的电路希望 X_2 比 X_1 大。

单元的形状影响单元在逻辑链中合适地放在一起，决定了比较复杂的单元看上去会像什么样子。把单元一块块拼在一起称为"铺瓦"，因为单元本身看上去像是不一致的瓦片。图 5.31(a)显示由四个瓦片构成的一个简单的逻辑链，它具有较大的 D 值。这使整个单元组合比起具有较小 D 值的图 5.31(b)来相对较短。在这种情况下，组合是短的，但相当宽。

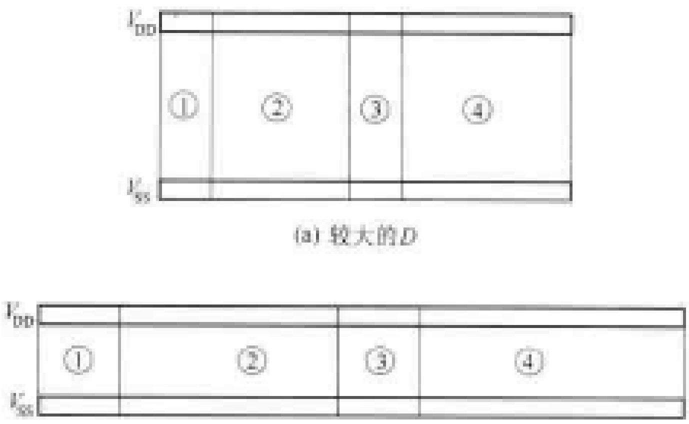
互连线布线对 $V_{DD}-V_{SS}$间距而言也是一个很重要的考虑。在复杂的数字系统中，布线常常比设计管子阵列更复杂。解决这个问题的一个方法就是把一排排的逻辑单元平行

放置，且在这些行之间留有空间用于布线。图 5.32 画出了这个一般的概念。与一排排逻辑平行走线的 Metal 1 线可以用来按需要布置信号线。由于 Metal 2 线可以横越过 Metal 1，可用垂直线把逻辑单元连接到 Metal 1 连线上。这个技术在 ASIC 设计中经常见到，因为它允许对不同的设计有相当程度的自由度。其主要的缺点是相对于紧密排布的版图，它的逻辑密度相对较低。



(a) 水平FET　　(b) 垂直FET

图 5.30　FET 方向对单元大小的影响



(a) 较大的D

(b) 较小的D

图 5.31　单元形状对组成大单元的影响

另一个高密度的技术是使 V_{DD}和 V_{SS}电源地线交替布置使在上面和下面的单元，共享这些电源线。这就是图 5.33 所示的"**Weinberger 图形**"。"倒置逻辑单元"是指它的方位与它上面或下面一行的"逻辑单元"的方位相颠倒，它的 V_{SS}在顶部而 V_{DD}在底部。在 Weinberger 镜像中 FET 放置的细节见图 5.34。nWell 区在 V_{DD}线周围因此可以在电源线的上部和下部构建 pFET。nFET 则放置在 V_{SS}线的两侧。因为不需要保留布线空间，所以这种办法可以高密度地放置单元。其主要缺点是在行之间的连接必须依靠 Metal 2 或更高的金属层，因为 Metal 1 已指派用于电源线。如果在一行的内部有足够的地方，那么也可以用水平方向的 Metal 1 作为一行内部的互连线。

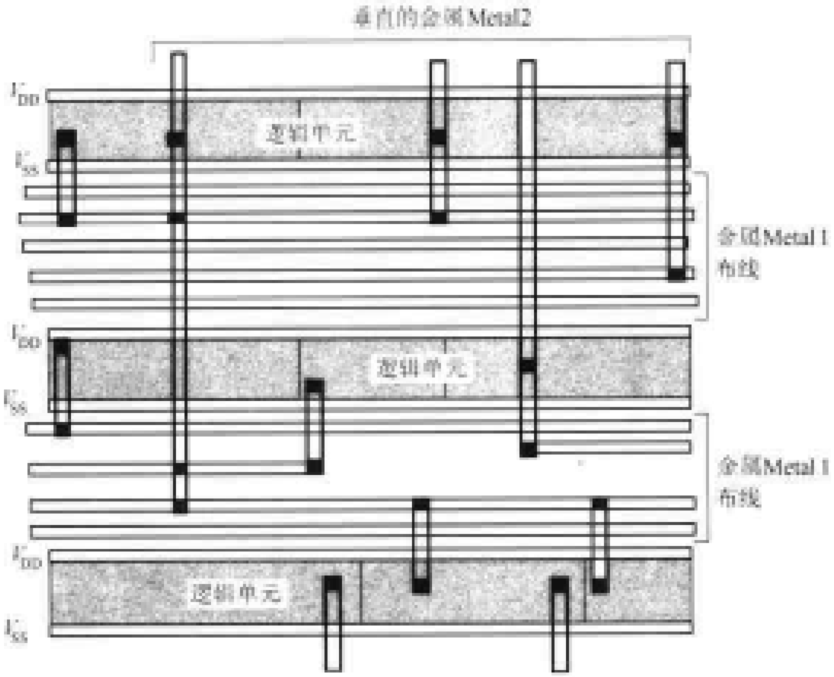


图 5.32 布线通道

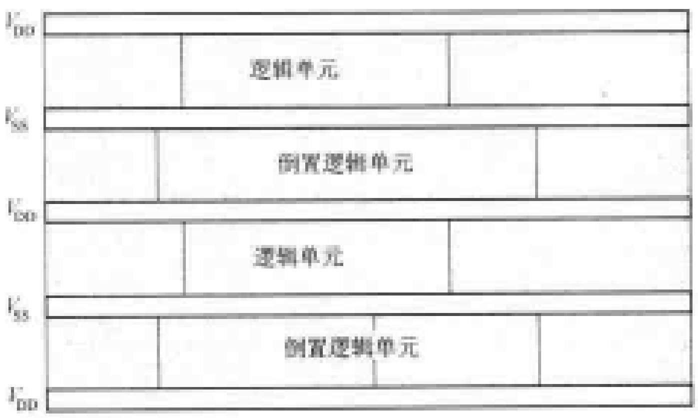


图 5.33 Weinberger 的镜像阵列

端口放置

一个单元的输入和输出端口必须放置在易于互连线布线的地方。在最初一层上，我们把逻辑电路的输入看成是 MOSFET 栅端口，而输出则是金属互连线。因为 FET 的栅是处在多晶硅层，我们必须提供一个多晶接触把一个单元的输出连至另一个单元的输入。

图 5.35 表示端口沿一个单元周边布置的情形。在这一简图中，输出多晶线在左边并且包括一个 Metal 1 的接线块和多晶接触。右边的输出则处在 Metal 1 层上，这可以使单元的互连在同一层上完成。图中也显示了垂直的多晶输入。如果版图运用图 5.32 所示的单元行之间布线通道的话，这些都是很有用的。

在布置单元端口时没有任何预先规定的限制，而且在实际中也可采用内部端口。最重要的是保证在复杂设计中，单元可以按要求用导线连接在一起。布线问题往往出现在关键的时

刻。仔细地布置单元和可靠的CAD工具,可以更有效地解决这些问题。

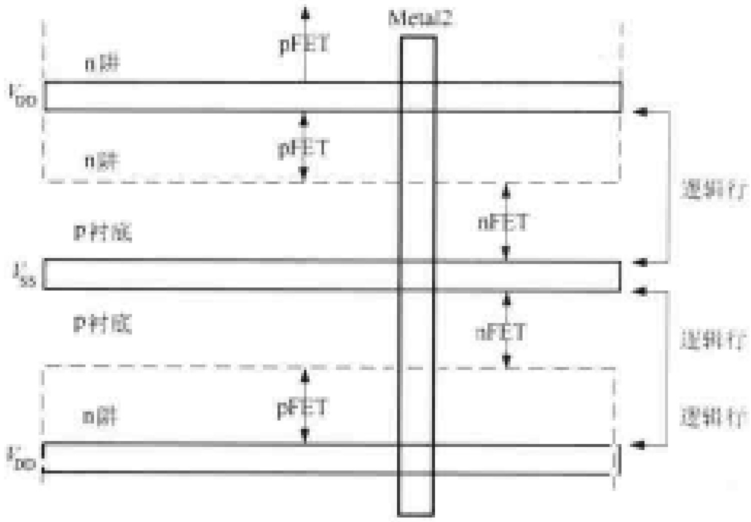


图5.34 Weinberger 阵列中FET的布置

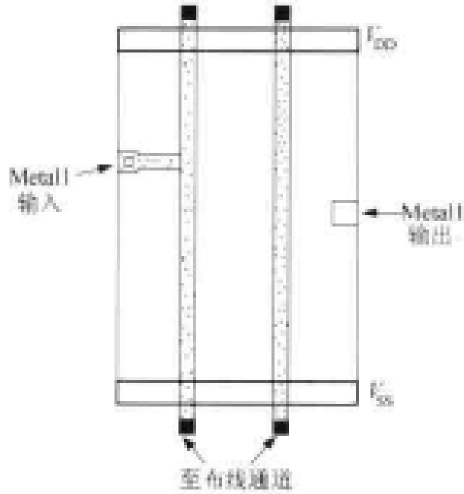


图5.35 单元中端口的布置

既然学习了逻辑单元的基本知识,让我们来学习在硅层上设计一组CMOS门的细节。一旦有了合理的一组门,便可进入下一个设计层次,即构建更为复杂的单元。

5.4 FET的尺寸确定和单位晶体管

场效应管是由宽长比(W/L)来说明的,这里W是沟道宽度而L是沟道长度。在现代VLSI中,沟长和沟宽为微米[μm]数量级,具体数值是在掩模的版图中确定的。这些尺寸连同工艺参数决定了管子的电气特性。

考虑图5.36中画出的基本FET。图中明确表示所画沟长和沟宽的尺寸。可以运用几个简单的公式,估计某些与版图有关的管子的电特性。首先栅的面积定义为覆盖在沟道区的多晶部分。图中显示了栅的面积$A_G = LW$。因此从栅端口(图中用G来标注)看进去的栅电容C_G就是:

$$C_G = C_{ox} WL \tag{5.19}$$

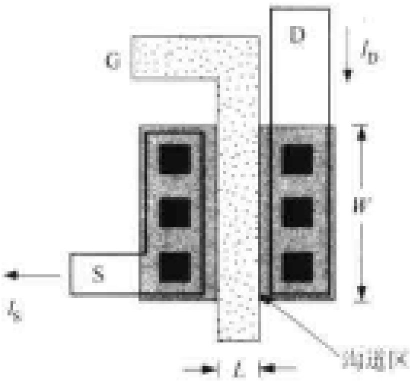


图5.36 FET的基本几何尺寸

我们可以回想起式中的C_{ox}是每单位面积的氧化层电容。

让我们来考察一下从漏(图中的D)至源(标记为S)通过器件的电流。进入漏端的电流表示为I_D而从源端流出的电流为I_S,于是

$$I_D \approx I_S \tag{5.20}$$

这是一个合理的近似。这就是说,电流是通过处在栅下面的沟道区从漏流向源的。沟道区本身具有电阻R_{chan}[Ω],它阻止电流的流动。如果把沟道模拟成一个简单的矩形块,那么电阻就可以近似为

$$R_{chan} = R_{s,c}\left(\frac{L}{W}\right) \quad (5.21)$$

式中 $R_{s,c}$为沟道区的薄层电阻。可惜的是 FET 并不那么简单，因此计算漏至源的电阻也是比较复杂的。然而这一方程与比较严格的分析是一致的，即它预见到 R_{chan}反比于沟宽 W：

$$R_{chan} \propto \frac{1}{W} \quad (5.22)$$

这就是说，增加 W 可以减小电阻，从而允许较大的电流。因此沟道尺寸确定了 FET 的电阻和电容。

另一个特点也是值得提及的。在 nFET 和 pFET 之间的最基本差别是产生电流的电荷极性。一个 nFET 运用的是带负电荷的电子，而一个 pFET 则依靠带正电荷的空穴。然而回想一下电子比空穴更容易移动，这可以由以下的关系来表示：

$$\mu_n > \mu_p \quad (5.23)$$

这在第 3 章 3.2 节中介绍过。在这一方程中，μ_n 和 μ_p 分别是电子和空穴的迁移率。迁移率值大的粒子意味着比迁移率小的粒子"更能移动"。假设我们设计具有同样宽长比(W/L)的 nFET 和 pFET，因为电子具有较大的迁移率，nFET 的电阻 R_n 就会小于 pFET 的电阻 R_p。让我们定义**迁移率的比** r 为

$$r = \frac{\mu_n}{\mu_p} \quad (5.24)$$

在现代 CMOS 工艺中，迁移率的比 $r>1$，通常在 2 和 3 之间，确切的值是由掺杂浓度和其他物理因素确定的。电阻与电导成反比，而电导则与迁移率成正比。因此可以得出结论：对于相同尺寸的 FET，R_n 和 R_p 的关系为

$$\frac{R_p}{R_n} = r \quad (5.25)$$

这在文献中常常叙述为 pFET 的导电不如 nFET 的好。换句话说，由于电子比空穴运动得快，由此得出结论 nFET 要比 pFET 快。这两个结论都假设进行比较的管子具有相同的尺寸。

FET 电阻可以通过改变沟道宽度 W 来调整。假设有一个 nFET 其宽长比为$(W/L)_n$，电阻为 R_n。为了设计一个 pFET，使它具有与 nFET 相同的电阻值 $R_p = R_n$，可以采用宽长比$(W/L)_p > (W/L)_n$，以补偿在迁移率方面的差别。这是通过选择下式来实现的：

$$\left(\frac{W}{L}\right)_p = r\left(\frac{W}{L}\right)_n \quad (5.26)$$

在这一设计中，电阻是相同的。然而注意到由于加大了 pFET 的沟宽，所以栅的面积是不同的，$A_{Gp} > A_{Gn}$。假设沟道长度相同，那么就会有不同的栅电容：

$$C_{Gp} = rC_{Gn} \quad (5.27)$$

因为面积正比于 W。

【例 5.1】 考虑一个宽长比$(W/L)_n = 4$ 的 nFET 用 $r = 2.4$ 的工艺制造。为了构造一个具有相同电阻的 pFET，必须选择

$$\left(\frac{W}{L}\right)_p = 2.4(4) = 9.6 \tag{5.28}$$

在实际中采用最接近的整数值 $(W/L)_p = 10$。pFET 的栅电容因此也会比 nFET 的栅电容大相同的倍数：

$$C_{Gp} = rC_{Gn} \tag{5.29}$$

一个同样值得提及的明显事实是，pFET 比 nFET 占有更大的表面面积。

管子的电气特性决定了 VLSI 电路的开关速度。在物理层，这相当于对电路中的每个 FET 选择它们的宽长比 $(W/L)_n$ 和 $(W/L)_p$。一旦确定了尺寸，物理设计问题也就成了用所说明的宽长比来设计硅电路。让我们集中在物理设计问题上。本书的其余许多章节集中在如何选择管子的尺寸，设计高速逻辑电路。

电路版图的一个很有用的起点是定义一个**单位晶体管**。这是一个具有规定的宽长比 (W/L) 的 FET，它可以按要求在版图上进行复制。因为它只需要画一次，因此比起设计者必须构建每个管子来，版图的完成可以快得多。而且由于器件的电气特性是已知的，因此开关性能的分析将是很直接的。

单位晶体管的一种选择是**最小尺寸的 MOSFET**。如其名字，一个最小尺寸的 FET 是运用设计规则设计的最小管子。图 5.37 就是一个例子。所画的沟道长度 L 是最小允许的多晶宽度 w_p，而所画的沟道宽度 W 是有源区掩模图形所允许的最小宽度 w_a。这个器件的宽长比因此为

$$\left(\frac{W}{L}\right)_{min} = \frac{w_a}{w_p} \tag{5.30}$$

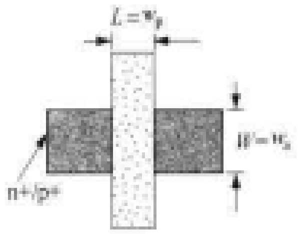


图 5.37　最小尺寸 FET 的几何尺寸

这通过观察就可以验证。栅电容则为

$$C_G = C_{ox} w_a w_p \tag{5.31}$$

这是因为栅面积即为 $A_G = w_a w_p$。最小尺寸器件是最小的晶体管，所以在理论上它可以得到最高的集成密度。然而它在任何 FET 中确实也具有最大的电阻，因此它可能并不是每个电路的最好选择。

图 5.37 所示的基本最小尺寸 FET 没有任何接触孔。如果增加有源区接触，以便用 Metal 1 连接，那么尺寸就会改变。考虑图 5.38(a) 中修改过的版图。沟长仍然为 $L = w_p$，然而由于在氧化层上采用了有源区接触刻孔，设计规则

d_c = 接触的尺寸

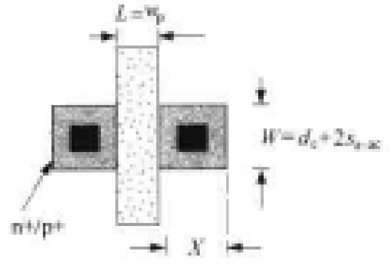
s_{a-ac} = 在有源区和有源区接触之间的间距

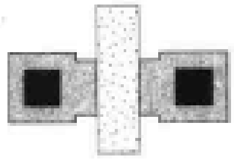
必须得到满足。最小的宽度为

$$W = d_c + 2s_{a\text{-}ac} \tag{5.32}$$

在某些工艺中，这一值可以与 $W = w_a$ 相同。如果不同，那么有源区可以放大以包含接触孔，如图 5.38(b) 所示。这使我们可以有 $W = w_a < d_c + 2s_{a-ac}$。尽管最小尺寸的 FET 由于它们

的高电阻而速度很慢,但在慢的开关速度不是一个主要考虑的地方仍然可能很有用。





(a) 有源区接触 (b) 小的 w_g

图 5.38 带有有源区接触的最小尺寸 FET

一旦选择了一个单位 FET,允许它的尺寸可以**放大**是很有用的。在图 5.39 中一倍(1*X*)的管子用来作为参照的基础,较大的管子则通过加倍其宽度来得到。图中显示了 2 倍(2*X*)和 4 倍(4*X*)管子的例子。改变管子的尺寸会改变它的电阻和电容。让我们分别用 R_{1X}和 C_{1X}来表示 1 倍器件的电阻和栅电容。如果一倍器件的宽度为 W_{1X},那么可以采用一个**放大比例因子** $S \geqslant 1$ 来构造较大的 FET,即

$$W_{SX} = SW_{1X} \tag{5.33}$$

例如,使 $S = 4$,就有

$$W_{4X} = 4W_{1X} \tag{5.34}$$

它描述一个 4 倍尺寸的管子。一个尺寸被放大的 FET 的电阻和电容的改变是因为电阻和电容都取决于器件的尺寸。应用放大变换可以得到一般化的值:

$$R_{SX} = \frac{R_{1X}}{S} \qquad C_{SX} = SC_{1X} \tag{5.35}$$

例如,对 2 倍的 FET 有

$$R_{2X} = \frac{R_{1X}}{2} \qquad C_{2X} = 2C_{1X} \tag{5.36}$$

这些很容易记忆。因为 pFET 的导电特性不同于 nFET,因此通常对每种类型 FET 都引入一个单位晶体管。但无论极性如何,放大关系仍维持不变。

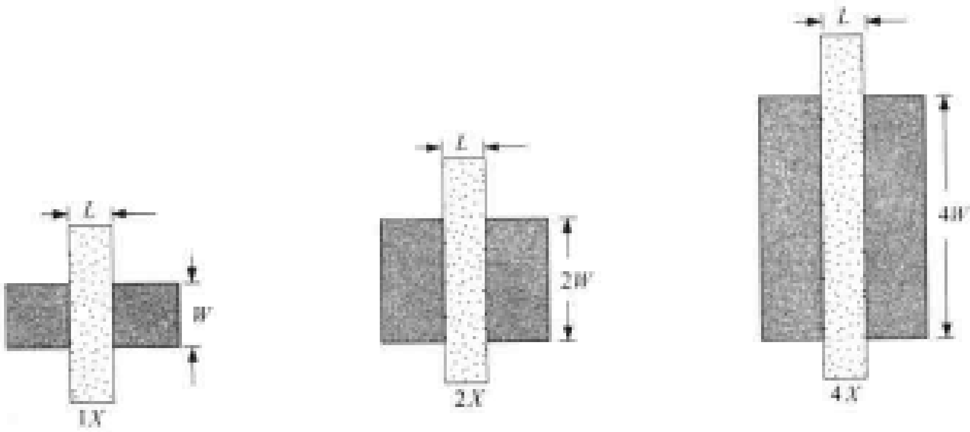


图 5.39 单位晶体管的放大

单位器件并不局限于单个晶体管。定义一个 FET 的串联和并联组合作为 $1X$ 的单元也是很有用的，然后就可以运用相同的技术对它进行放大。图 5.40 表示尺寸分别为 1 倍和 2 倍的两个 FET 的串联链。因为每个管子都是以相同的方式扩大的，因此电阻和电容的关系仍然成立。然而应当注意到，串联管子的总电阻是各个管子电阻的和。如果一个 1 倍 FET 的电阻为 $1R_{1X}$，那么串联组合的电阻就是 $2R_{1X}$。由于在 2 倍尺寸的电路中每个 FET 的电阻为 $(R_{1X}/2)$，因此被扩大的 2 倍管子串联组合的电阻只要相加就可以得到

$$2\,(R_{1X}/2) = R_{1X} \tag{5.37}$$

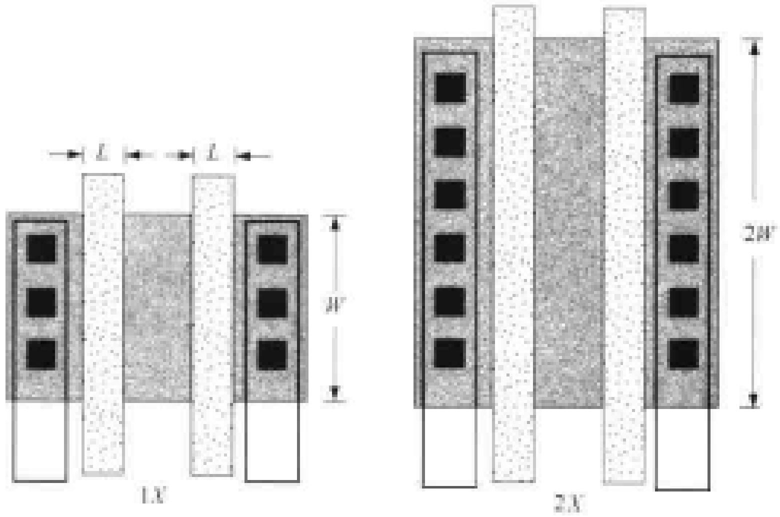


图 5.40　串联连接 FET 链的放大

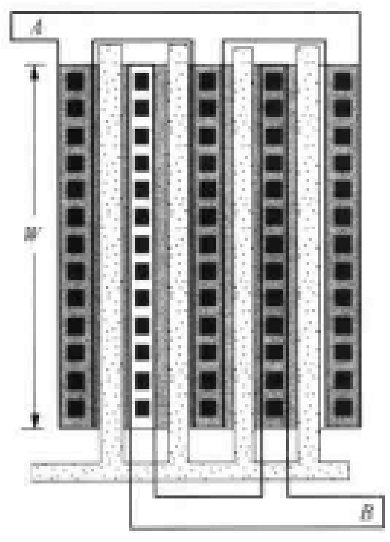


图 5.41　用平行连接的管子构成较大的 FET

串联 FET 通常都做得比单个管子更大，以降低端至端的总电阻。

大的晶体管常常要求考虑更多一点，有时出现宽长比达到 100 或更大的情形。具有较大沟宽 W 的单个器件将具有很长的矩形形状，因而有可能很难放进整个版图中。或者栅材料的电阻会使信号速度变慢。

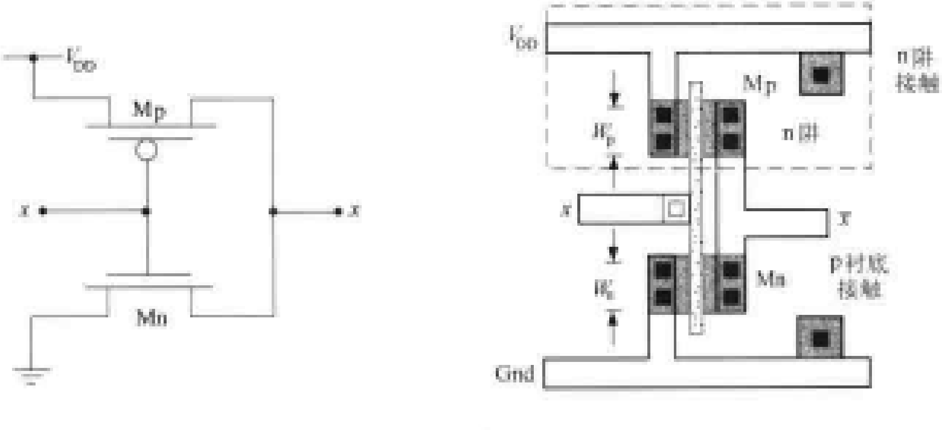
最常用的解决办法是采用一组并行连接的管子。图 5.41 显示了以宽度 W 为基础的一组管子。四条栅线都连接在一起，导线布线则使 A 边和 B 边之间的等效沟道宽度为 $4\,W$。这个方法的一个优点是整个版图的几何形状可被调整成方形或接近方形。

5.5　逻辑门的物理设计

让我们把物理设计过程的基本知识，用来构建一组基本的 CMOS 逻辑门的版图。每个门被归类为一个单独的单元。我们将集中在单位单元的设计，因为大的单元可以通过放大尺寸来得到。

5.5.1 NOT 单元

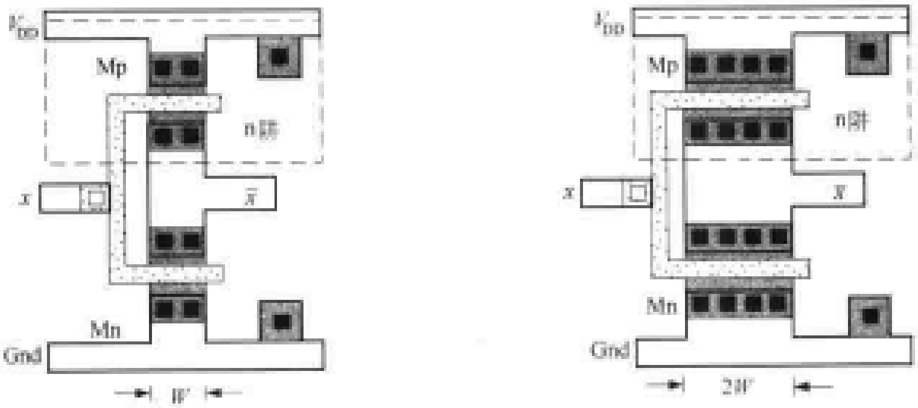
最简单的 CMOS 逻辑电路是实现 NOT 操作的反相器。考虑图 5.42(a)所示的电路图，水平方向定位可以直接转换成如图 5.42(b)所示的版图。版图中显示 FET 的沟宽 W_p 和 W_n。图中也显示了 V_{DD}至 n 阱(它是 pFET 的衬底)和 Gnd 至 p 衬底(它是 nFET 的衬底)的重要连接。这些连接在我们的图中并不总是明确显示，但却必须包括在每个单元中以得到功能正确的电路。



(a) 电路图　　(b) 单元版图

图 5.42　FET 水平布置的非门

尽管这一 FET 非门的简单例子说明了版图的基本特点，但 V_{DD} − Gnd 的 Metal 1 线的较小间距却使它很难扩大。如果我们把 FET 旋转 90°，那么就很容易增加 FET 的沟道宽度。这显示在图 5.43 的例子中。在图 5.43(a)中，单位 NOT 的设计对 Mn 和 Mp 都具有相同的宽长比(W/L)。图 5.43(b)中 2 倍的单元运用相同的 V_{DD} − Gnd 节距，但通过把 FET 沿水平方向扩展就得到了宽长比为 2(W/L)的管子。



(a) 基本单元　　(b) 2倍单元

图 5.43　FET 垂直布置的 NOT 门版图

另一个例子显示在图 5.44 中。在这一设计中，根据迁移率 pFET 应比 nFET 大 $r = 2.5$

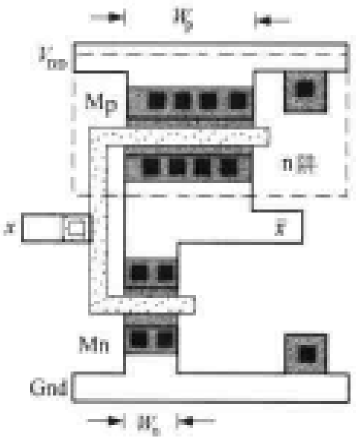


图 5.44　电气对称的非门版图

倍。这就使在输出 x̄及两条电源线 V_{DD}和 Gnd 之间的电阻相等。由于 nFET 和 pFET 具有相同的电阻,这个反相器称为**对称反相器**(尽管它在几何尺寸上并不对称)。这一设计的意义将在第 7 章研究 CMOS 逻辑门的电气设计时讨论。

5.5.2　与非门(NAND)和或非门(NOR)单元

可以采用相同的技术来设计与非门(NAND)的版图。图 5.45(a)的 NAND2 版图采用垂直放置的 FET。在这一设计中,所有的管子都具有相同的宽长比,它们可以按需要重新确定尺寸,整个单元本身也是如此。如果运用了更多的输入,如在 NAND3 门中那样,那么确定 nFET 的尺寸变得尤为关键。在这种情形中,W_n 的值应当增加,以减少从输出至接地端的串联电阻。

一个 NOR 门也可以用同样的方式构造。图 5.45(b)中的 NOR2 版图可以通过翻转 NAND 的版图并且重新定义 FET 的极性和电源线就可以得到。这一设计也采用尺寸全部相同的 FET。然而由于 pFET 具有相对较高的电阻值,从输出至 V_{DD}的串联 pFET 链会引起过长的开关延时。采用较大的 W_p 值可以使这一延时缩短。

另一种 NAND2 和 NOR2 门的版图显示在图 5.46 中,它们采用垂直走向的栅图形;这个方法的布线已在第 3 章中研究过。然而这里的图更为详细地显示 FET 的尺寸。这两个版图对串联的管子都加大了沟道宽度以降低电阻。在图 5.46(a)中 NAND2 门的 nFET 设计得比在图 5.46(b)中 NOR2 门的并联 nFET 要宽些。类似地在 NOR2 中串联的 pFET 比在 NAND2 中并联的 pFET 要宽些。

W_n 和 W_p 的实际数值决定了一个门的电气特性。在许多设计中,版图采用易于确定尺寸的 FET,然后对电路进行模拟以确定它们的电气响应,如果需要则调整尺寸。在关键的数据路径上,这些值更为重要,因此最初的设计工作集中在找到可接受的值。在本书后续章节将讨论逻辑门设计电气方面的内容。

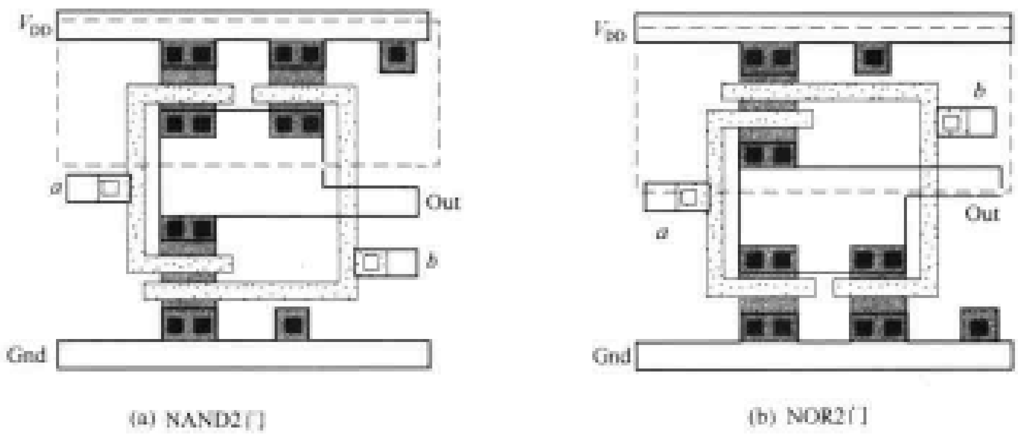


图 5.45　FET 垂直放置的 NAND2 和 NOR2 版图

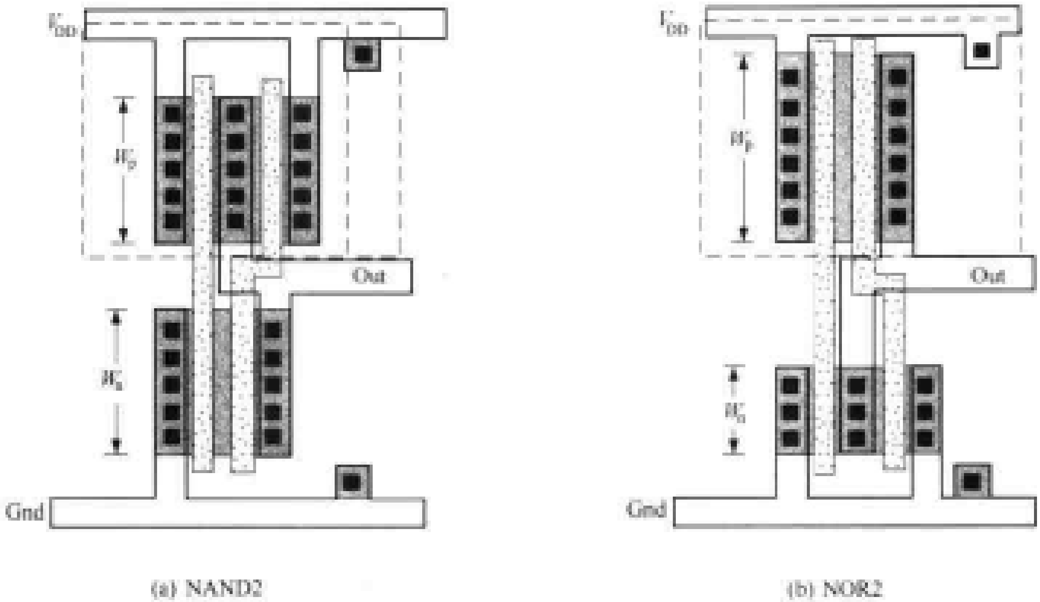


图 5.46　NAND2 和 NOR2 单元的另一种版图

5.5.3 复合逻辑门

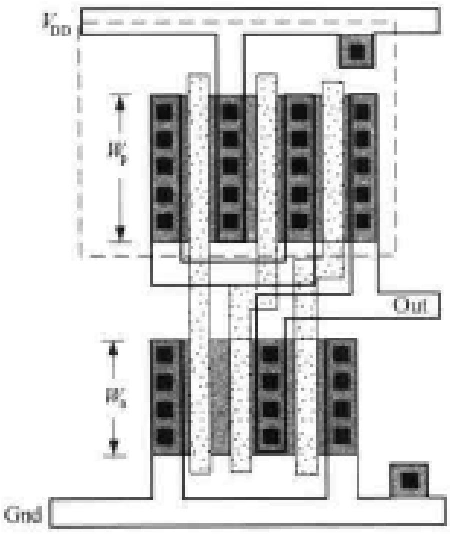


图 5.47　复合逻辑门的例子

复合逻辑门可用同样的方式进行版图设计。第 3 章中介绍的布线技术可以用来布置器件。在物理设计阶段，每个管子的尺寸被确定，同时 FET 结构被放在 V_{DD}和 Gnd 线之间。串联的 FET 一般都要设计得比单个的管子宽些，除非它们共享同一个有源区或者出于其他重要的考虑。

图 5.47 为一个复合逻辑门的例子。由于 nFET 和 pFET 阵列共享漏/源区域，因此 W_n 和 W_p 采用同一个值以简化版图。注意在给定的 V_{DD} - Gnd 间距内可使 pFET 宽些以补偿它们较高的电阻值。

5.5.4 关于版图的小结

以上这些例子提供了运用以下顺序对逻辑门进行物理设计的基础知识：

- 设计 MOSFET 逻辑电路
- 运用晶体管电路建立布线图，图中只有布线路径和层次是重要的
- 运用布线图作为该门最终物理设计的基础，它包括所有图形的合适尺寸并符合设计规则

逻辑门设计的最后一个方面对于初学者来说可能是相当费时的。某些复杂的电路即使对于一个有经验的设计者来说也是很难的。在一个版图完成后，它必须被提取并用工艺参数进

行模拟。一个单元是否最终被接受，取决于它是否满足所有的电气和尺寸方面的要求。

VLSI设计者总想在一个给定的面积上放置尽可能多的电路。在工程设计层次上这是通过运用规则和重复图案的概念以及掌握CAD工具来实现的。从设计简单单元中学到的内容，可为设计日益复杂的电路打下基础。设计自动化工具的功能正在变得非常强和更智能化，它们正在为难以置信的复杂设计铺平道路。

5.6 设计层次化

VLSI系统是采用设计层次化的概念构成的，即采用简单的建筑块来设计比较复杂的单元。这一嵌套一直继续到整个芯片完成时为止。一个版图编辑器的程序被编写来为芯片设计者提供这类环境。建立层次化设计的关键依赖于单元的概念。我们定义一个单元是一些对象的集合，它被看成单个的实体。对象本身的这些特征提供了层次化的基础。

最简单的单元只包含多边形。像在前一节中NOT和NAND2例子这样的逻辑门就属于这一类。具有这个特性的单元称为**最底层**的单元(**flat**)，这是指每个对象都是独立的并且与任何其他的对象无关。在最底层单元中，可以改变任何多边形而不会影响任何其他的多边形。为了开始设计过程，需要建立大量的底层单元且把它们存放到库中去。最基本的库单元实体是晶体管以及在较复杂设计中用做建筑块的逻辑门。图5.48表示了这一概念。在这一简单的例子中，在多边形这个层次上构造3个门级设计nor2，nand2以及not，然后每个被存放起来作为库中的独立单元。每个单元与其他单元无关。

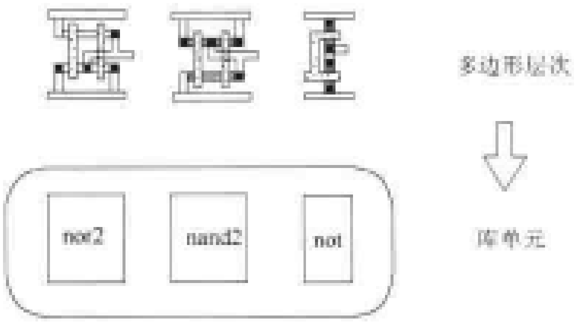


图5.48 最基本的多边形层次的库单元

一旦建立最初的库，可在设计中运用这些单元实体，即把它们例举到版图中。一个例图是库中单元的复制品。一个被例举的对象在新的版图中是不可以改动的，因为它总是库单元实体的确切复制。改变一个例图特征的惟一办法是改变库单元实体。需要掌握的最重要的概念是这个新的版图将是一个更复杂的对象，它本身又可作为一个单元存放在库中。在图5.49中，两个新单元cell _ 1和cell _ 2的设计采用原始库的例图加上它们自己的多边形。可把这个新单元存放起来并且组合一个更大的库(Library 1)，用在更为复杂的设计中。这个过程可按需要进行重复。把有用的功能设计成新的单元，成为库的一部分并且用来建造其他单元。为库选择的最终单元集合应当包含设计项目所需要的最大多数单元。

单元层次化的概念基于单元库的建立。图5.50显示了这一策略。在最初始的层次上，单元只是由代表材料层的多边形构成。这在层次化中表示为Level _ 1。Level _ 2的单元由多边形以及Level _ 1单元的例图构成。再往上一组则表示Level _ 3的单元。它们包含多边形，也可以包含Level _ 1和Level _ 2实体的例图。图中显示的最后一组是Level _ 4的单元，它们由多边形以及从Level _ 1至Level _ 3中所有单元的例图构成。

应当记住，一个例图只是一个较为简单实体的复制，它内部的结构不能在较高的层次上改变。例如，Level _ 2的单元被例举到Level _ 4，那么在Level _ 4上的设计把它看成不可改动的。为了改动这个Level _ 2的单元，设计者必须回到原先的Level _ 2设计。在这个单元中的

任何改变都会传递给这个单元被例举到的所有较高的层次。实际上库为大量设计者使用，但大多数的用户却没有进入核心单元组的特权。这就防止有人去改变对另一个人的设计，可能是很关键的特性。

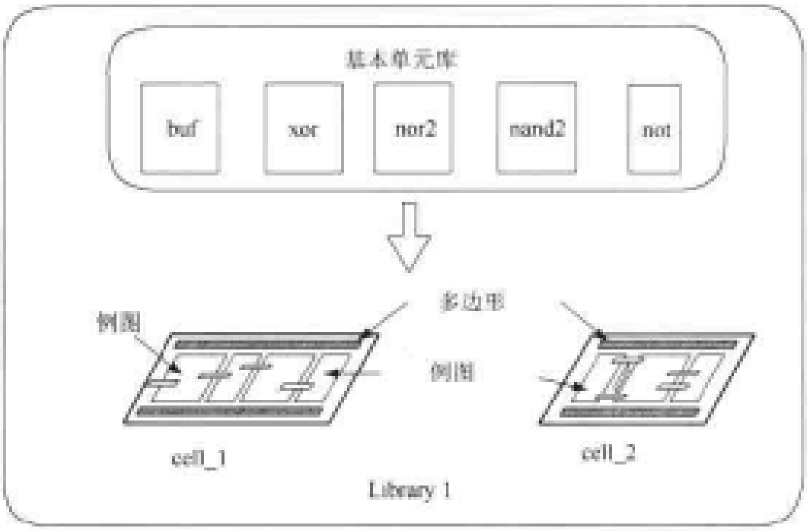


图 5.49 用更复杂的单元扩充库

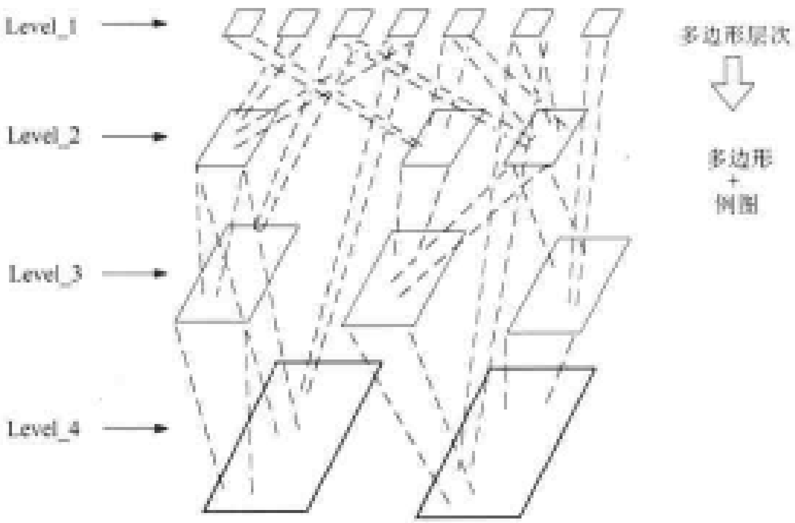


图 5.50 单元层次

尽管一个例图的内容不可改变，却有可能通过"展平"(flatten)的命令把它分解成许多多边形。当一个单元被"展平"后，对原始单元的所有参照关系都将失去，因而这个电路的各个图形都可以被修改。图 5.51 显示了"展平"操作的影响。一个被"展平"的单元不能再恢复成它原先被例举的形式。

设计层次化的概念在 VLSI 工程中是不可缺少的。它可以从最基本的层次上开始并且加入认为有用的单元，从而可以构造复杂的电路。各种各样的库以这一方式都可以建立起来和进行维护，以便用在许多不同的项目中。复杂的系统可被分解成易于处理的几部分，从而使以成千上万个管子来设计芯片的想法成为现实。

总之,版图是与工艺相关的。这意味着每当一个新的生产厂启动时必须建立一个新的库。除非新的工艺是根本不同的,否则老的单元也可用来作为新库的起点。常有可能只是缩放尺寸,这就是所谓单元**复用**概念的基础。它可以帮助缩短制造新芯片所需要的时间。当前的许多设计都是在考虑复用的情况下完成的。

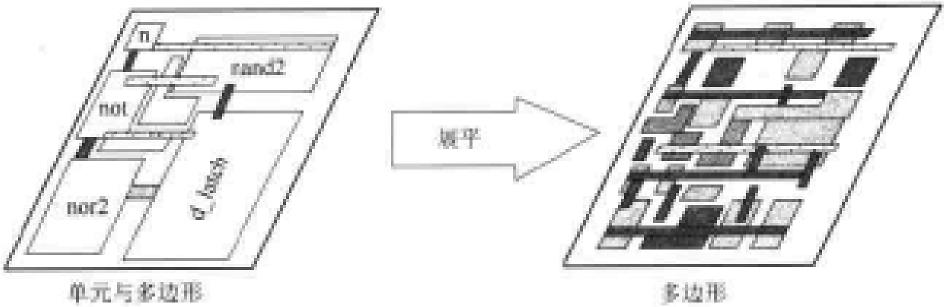


图 5.51 展平操作的影响

5.7 参考资料

[1] R. Jacob Baker, Harry W. Li, and David E. Boyce **CMOS Circuit Design, Layout and Simulation**, IEEE Press, Piscataway, NJ, 1998.

[2] H. B. Bakoglu, **Circuits, Interconnections, and Packaging for VLSI**, Addison-Wesley, Reading, MA, 1990.

[3] Kerry Bernstein, et al., **High-Speed CMOS Design Styles**, Kluwer Academic Publishers, Norwell, MA, 1998.

[4] Dan Clein, **CMOS IC Layout**, Newnes Publishing Co., Woburn, MA, 2000.

[5] Robert F. Pierret, **Semiconductor Device Fundamentals**, Addison-Wesley, Reading, MA, 1996.

[6] Bryan Preas and Michael Lorenzetti (eds.), **Physical Design Automation of VLSI Systems**, Benjamin/Cummings Publishing Co., Menlo Park, CA, 1988.

[7] M. Sarrafzadeh and C.K. Wong, **An Introduction to VLSI Physical Design**, McGraw-Hill, New York, 1996.

[8] Jasprit Singh, **Semiconductor Devices**, John Wiley & Sons, New York, 2001.

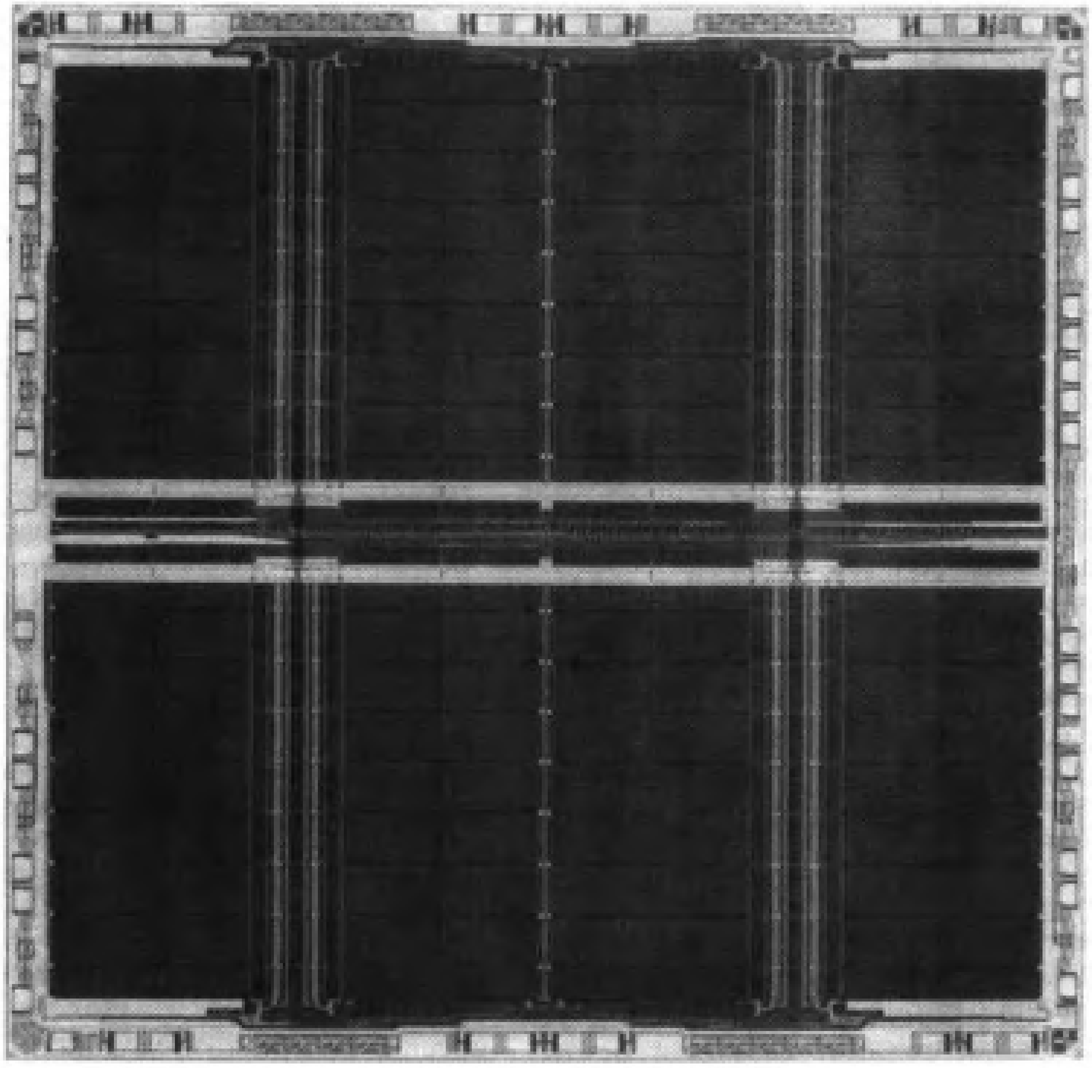
[9] Ben G. Streetman and Sanhay Banerjee, **Solid State Electronic Devices**, 5th ed., Prentice Hall, Upper Saddle River, NJ, 1998.

[10] R. R. Troutman, **Latchup in CMOS Technology**, Kluwer Academic Publishers, Norwell, MA, 1986.

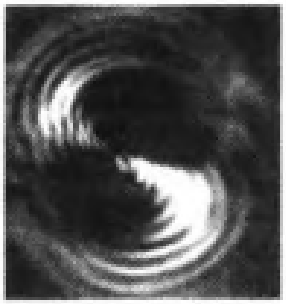
[11] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[12] John P. Uyemura, **Physical Design of CMOS Integrated Circuits Using L-Edit™**, PWS Publishing Company, Boston, 1995.

[13] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.



第 2 部分



从逻辑到电子电路

第 6 章　MOSFET 的电气特性

本章中心是 MOSFET 的特性，也是学习 VLSI 电子学部分的开始。在这部分电流和电压是最重要的物理量，但本章重点不在于为电子学本身而学习电子学，而是侧重于物理设计与逻辑电路之间的联系。

6.1　MOS 物理学

MOSFET 通过外加电压使电荷从器件的源端移动到漏端来传导电流。由于漏和源在物理上是分开的，栅下的电荷移动只有在建立了导电通道，即**沟道**时才会发生。

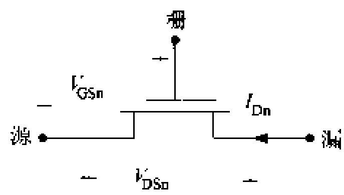


图 6.1　nFET 的电流和电压

考虑图 6.1 中 nFET 的电路图符号。漏电流 I_{Dn}由器件上的外加电压控制。图中最基本的电压为栅—源电压 V_{GSn}和漏—源电压 V_{DSn}。为了得到器件的工作模型，首先要确定电流与电压(I-V)的关系

$$I_{Dn} = I_{Dn}(V_{GSn}, V_{DSn}) \tag{6.1}$$

一旦有了这一关系，就具备了分析和设计 CMOS 开关电路所需要的物理知识和数学模型。

我们学习的起点是图 6.2 简单的 MOS 结构；回忆一下缩写词“MOS”是用来描述任意的导体—氧化层—半导体的叠层，即便最上面的一层并不一定是金属。就图中的情况，最上面的导电层是栅层。该图代表的是一个 nFET 的中心区域，并且表现了导电层是如何在漏与源区之间形成的物理过程。栅上的外加电压用 V_G 表示，极性如图所示并假定为正值。氧化层是二氧化硅(SiO_2)，它在栅与衬底之间起绝缘层的作用。于是得到单位面积的氧化层电容为

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.2}$$

式中 t_{ox}为氧化层厚度，单位为 cm。回想在第 3 章中曾提到二氧化硅的介电常数为 $\varepsilon_{ox} = 3.9\varepsilon_0$，这里 $\varepsilon_0 = 8.854\times10^{-14}$F/cm 是真空的介电常数。现代 CMOS 工艺中的氧化层非常薄，典型值为 $t_{ox} < 10$ nm $= 10^{-6}$ cm。

C_{ox}的值决定了在栅电极和 p 型硅区之间的电耦合的程度。其效应在**硅表面**即硅区的最上面最为明显。电耦合用电场 E(单位为 V/cm)来描述，它是在电压加到栅上时在绝缘氧化层中形成的。电场在半导体中感应电荷，可以通过改变栅电压 V_G 来控制通过 FET 的电流。这就是**场效应**这一术语的来源。

为了描述场效应，引入表面电荷密度 Q_S 这一概念，它的单位为库仑/平方厘米[C/cm^2]。它与栅电压的关系为

$$Q_S = -C_{ox}V_G \tag{6.3}$$

表面电荷的概念可以通过图 6.3 来理解。实际上，Q_S 代表从氧化层向下看半导体所看到的电荷密度。式中含负号是由于正电压 V_G 感应出负的表面电荷密度。这个公式看起来虽然十分简单，但由于 Q_S 代表了在半导体表面的所有电荷，而这些电荷的性质又取决于外加栅电压，所以 MOS 的物理过程是很复杂的。

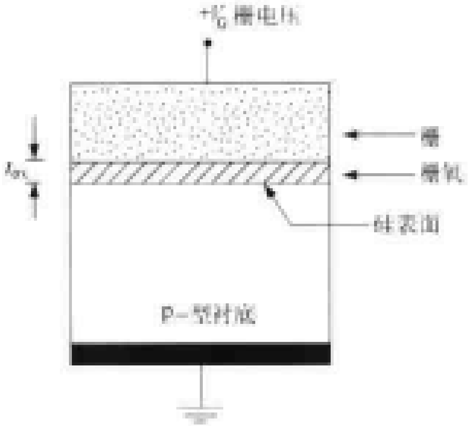


图 6.2 MOS 的结构

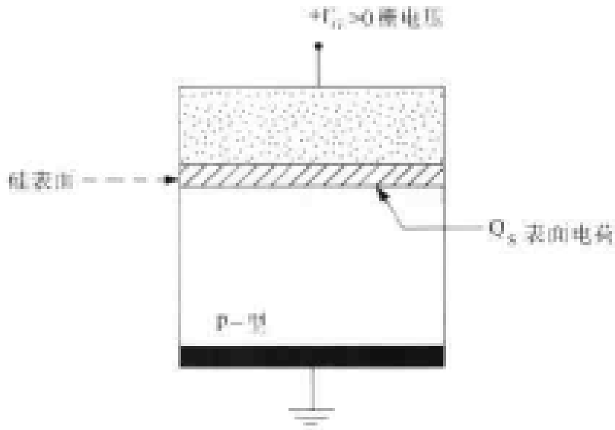


图 6.3 表面电荷密度 Q_S

在电路级上，将 Kirchhoff 电压定律(KVL)[①] 应用于图 6.4 中的 MOS 系统就可以得到阈值电压。假设栅电压 V_G 的极性如图所示，根据 KVL 定律得到表达式

$$V_G = V_{ox} + \phi_S \tag{6.4}$$

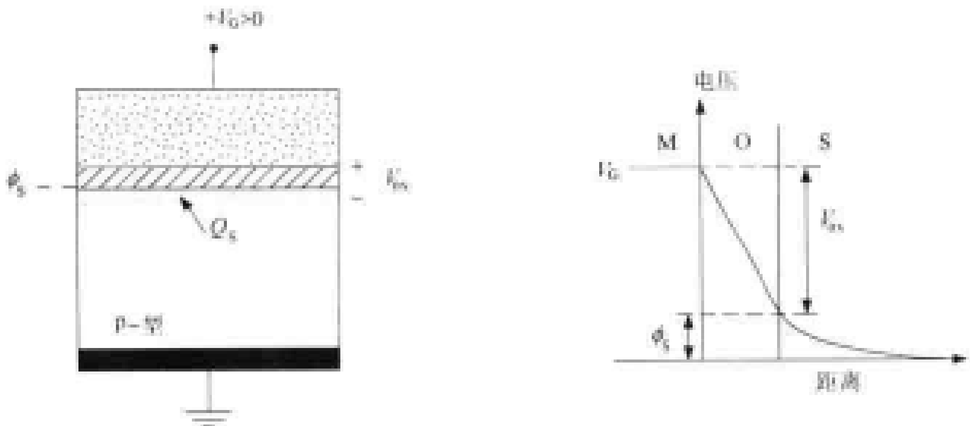


图 6.4 MOS 结构中的电压

式中 V_{ox}是通过氧化层的电压降，ϕ_S 是**表面电势**，它代表硅最上面的电压。MOS 系统的电压可画成曲线，V_G 为栅上电压，ϕ_S 位于硅表面。氧化层电压 V_{ox}为它们的差($V_G - \phi_S$)，是氧化层内电势下降的结果。同时注意，在半导体中的电压从 ϕ_S 值下降到 $\phi = 0$，趋势逐渐变缓。

图 6.5 是 MOS 系统中电场的情况，将氧化层的垂直尺度放大，以便看得更详细些。可以看到，绝缘体中的**氧化层电场** E_{ox}的方向由较高电位的栅极离开向下。**表面电场** E_S 也指向同

① 该定律称，在一个闭路电路中，电压升的总和等于电压降的总和。

样方向(指向接地端),并且控制着半导体表面的表面电荷密度。这是由于电场对带电粒子存在力。根据 Lorentz 定律这个力为

$$\boldsymbol{F} = Q_{\text{particle}}\boldsymbol{E} \tag{6.5}$$

式中,Q_{particle}是粒子所带的电荷,带有适当的符号。带正电的空穴具有一个 $+q$ 的电荷,而电场力的公式为

$$\boldsymbol{F}_{h} = +q\boldsymbol{E} \tag{6.6}$$

它表明空穴受到一个与电场同样方向的力②。反之,电子带有一个负电荷 $-q$,所以它们受到的力为

$$\boldsymbol{F}_{e} = -q\boldsymbol{E} \tag{6.7}$$

这里,负号表示电子受到一个与电场方向相反的力。如图 6.5 所示,表面电场 E_S 的方向向下,所以正电荷被驱赶离开表面面负电荷则被吸引至表面。这就解释了表而电荷密度为何由负电荷构成,即 Q_S 本身是一个负值。

表面电荷的性质取决于外加栅电压的大小。假设 V_G 由 0 开始增加到一个较小的正值,比如 $V_G = 0.1$ V,表面电场将电子吸向表面并将空穴推向下,其结果是半导体的表面产生了负电荷,它称为**体电荷密度** $Q_B<0$,单位是 C/cm²。体电荷是由于在 p 型衬底中存在硼原子。由于硼的作用是一个受主,它能俘获并且保持一个带负电荷的电子。这时,它变成了带一个净负电荷的**电离了的掺杂剂**。由于这些离子不能移动,所以体电荷也不能移动。由物理分析得到

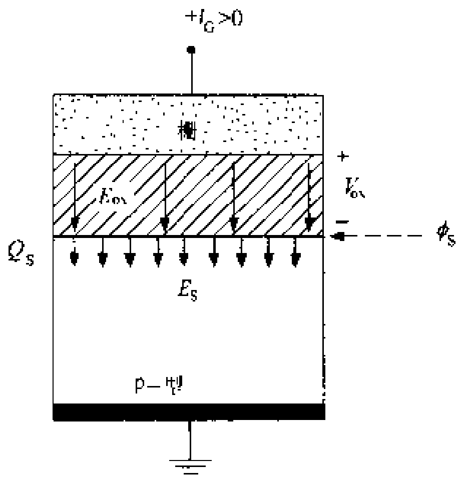


图 6.5　MOS 结构中的电场

$$Q_B = -\sqrt{2q\varepsilon_{Si}N_a\phi_S} \tag{6.8}$$

式中,ε_{Si}是硅的介电常数 $\varepsilon_{Si}\approx 11.8\varepsilon_o$。对这一情形,氧化层电压与体电荷的关系是

$$Q_B = -C_{ox}V_{ox} \tag{6.9}$$

图 6.6 显示了体电荷的情况,在图中用带负号的圆圈表示。从硅表面到**体**电荷层底部的部分叫**耗尽区**。因为它"耗尽"了自由电子和空穴:空穴被驱使离开,面电子则被硼掺杂原子"吸收"。耗尽区的厚度 x_d 随外加电压的增加面增加。在 MOS 系统中这种情况定义为"耗尽工作方式"。一个被耗尽的 MOS 结构不能支持电子电流的流动,因为体电荷已被硅晶格束缚不能移动。

如果将栅电压增加到一个特定的值,称为**阈值电压** V_{Tn},我们会观察到电荷性质的变化。阈值电压的名字就意味着它是两种不同现象的分界。当 $V_G< V_{Tn}$时,电荷为不能移动的体电荷,且 $Q_S = Q_B$。但是当 $V_G> V_{Tn}$时,电荷由两个不同的部分构成,即

$$Q_S = Q_B + Q_e < 0 \tag{6.10}$$

② 基本电荷的数值为 $q=1.6\times10^{-19}$库仑。

式中 Q_B 为体电荷。但现在我们观察到有一个电子电荷层，它用 Q_e(C/cm²)表示。图 6.7 是这两个部分的表面电荷。重要的是，电子是可以移动的，并且可以沿横向(平行于表面)运动。因此，可用该电子层作为沟道区建立一个 MOSFET。阈值电压 $V_G = V_{Tn}$ 代表了 Q_e 正好开始形成时的栅电压值。这就意味着当 $V_G = V_{Tn}$ 时 $Q_e = 0$。但是根据电容器关系式：

$$Q_e = -C_{ox}(V_G - V_{Tn}) \tag{6.11}$$

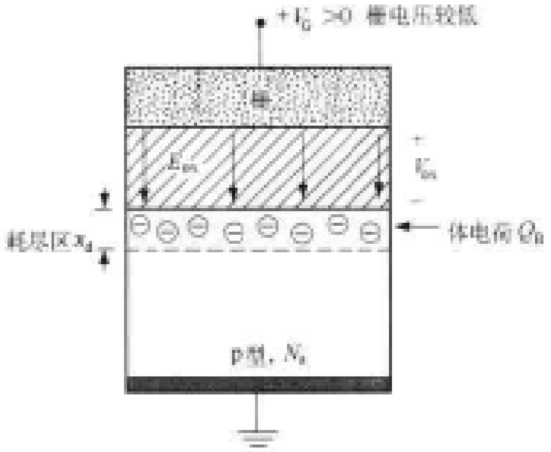


图 6.6 MOS 结构中的体(耗尽)电荷

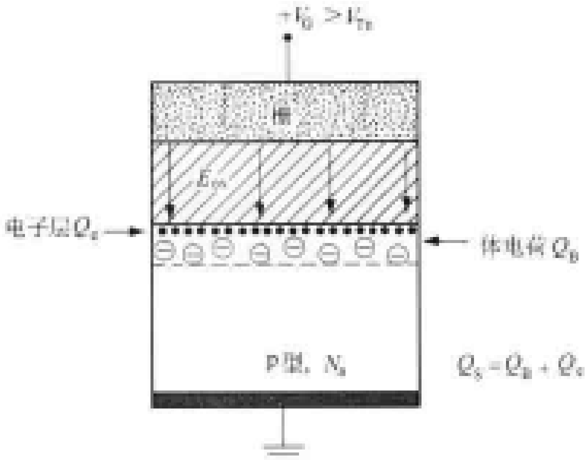


图 6.7 电子电荷的形成

可知 Q_e 在 $V_G > V_{Tn}$ 时会增加。我们必须从 V_G 中减去阈值电压以得到在电子层形成后绝缘体上的有效电压。注意，这意味着当栅电压满足 $V_G > V_{Tn}$ 时，体电荷 Q_B 并不会增加。式中负号表示电子带负电荷。

阈值电压的数值在生产过程中设定。典型值的范围大约为在 $V_{Tn} = 0.5$ V 到 $V_{Tn} = 0.8$ V之间，这取决于电路想要应用的类别。在 VLSI 系统设计中，假定 V_{Tn} 的值已在电气参数表中说明。

阈值电压的推导③

要得到一个说明阈值电压数值来源的近似公式并不困难。回想一下 KVL 电压公式

$$V_G = V_{ox} + \phi_S \tag{6.12}$$

对 MOS 系统的进一步研究表明，当表面电势达到

$$\phi_S = 2|\phi_F| \tag{6.13}$$

时，电子层正好开始形成。式中 $|\phi_F|$ 叫做**体费米电势**，它是由 p 型半导体中硼的掺杂密度 N_a 确定的。由分析得到

$$|\phi_F| = \left(\frac{kT}{q}\right)\ln\left(\frac{N_a}{n_i}\right) \tag{6.14}$$

式中 k 为 Boltzmann 常数，T 是绝对温度，单位为 Kelvin(K)。参数组合 (kT/q) 也称为**热电压** V_{th}，在室温($T = 27$ ℃ $= 300$ K)下其数值约为 $(kT/q) \approx 0.026$ V。

---

③ 本节可以跳过，而不会失去讨论的连续性。

建立上述关系后，可将 KVL 等式 $V_G = V_{Tn}$ 写成

$$V_{Tn} = V_{ox}\big|_{\phi_s = 2|\phi_F|} + 2|\phi_F| \tag{6.15}$$

运用上面求 Q_B 的公式(6.8)和(6.9)得到

$$V_{Tn} = \frac{1}{C_{ox}}\sqrt{2q\varepsilon_{Si}N_a(2|\phi_F|)} + 2|\phi_F| \tag{6.16}$$

这是一个**理想 MOS** 结构的阈值电压。在这一结构中，氧化层中没有任何寄生电荷，而且栅和半导体的材料是相同的。一个考虑更实际情况的一般表达式为

$$V_{Tn} = \frac{1}{C_{ox}}\sqrt{2q\varepsilon_{Si}N_a(2|\phi_F|)} + 2|\phi_F| + V_{FB} \tag{6.17}$$

式中，V_{FB} 叫做**平带电压**，它考虑了在氧化层及不同的栅和衬底材料中的电荷[④]。在最现代的 CMOS 工艺中，V_{FB} 是一个负数，这使 $V_{Tn}<0$。由于事实上大多数 CMOS 电路都是在正的电源电压下工作的，希望有一个正的阈值电压，即 $V_{Tn}>0$。这是通过引入另外一个工艺步骤来完成的，即将更多的硼离子注入这个区域的表面。这使阈值电压公式变为

$$V_{Tn} = \frac{1}{C_{ox}}\sqrt{2q\varepsilon_{Si}N_a(2|\phi_F|)} + 2|\phi_F| + V_{FB} + \frac{qD_I}{C_{ox}} \tag{6.18}$$

式中 D_I 为注入剂量，即每平方厘米注入的离子数；D_I 的单位为 cm^{-2}。所以，通过调整注入剂量可以设定阈值电压。在某些工艺中，也有可能通过改变栅的掺杂，改变平带电平 V_{FB}，从而改变阈值电压。

6.2 nFET 电流-电压方程

让我们将兴趣转向 n 沟道 MOSFET 的 I-V 特性上。这些是由器件本身的物理结构决定的。nFET 是由在一个 MOS 电容器的两边各加上一个 n+ 区构成的。图 6.8(a)的截面图显示源和漏的 n+ 区相对于 MOS(栅—氧化层—衬底)电容器是如何放置的。在 n+ 区两边之间的距离用符号 L 表示，称为器件的(电)**沟道长度**。L 具有长度单位，是 FET 中最小的特征尺寸。由于漏与源间的差别在加入外电压之前不能确定，所以此时它们的标记完全是随意的。作为以后的参考，注意到在一个 nFET 中，漏是具有较高电压的 n+ 一边。图 6.8(b)是 nFET 的顶视图。它确定了(电)**沟道宽度** W，单位也是长度。无量纲量(W/L)是宽长比，它用来说明一个晶体管相对于电路中其他管子的相对尺寸。

需要特别注意的是，本章中所用的 W 和 L 值是从电气上考虑的尺寸即"有效"值。而不是前一章介绍的版图设计时所画的尺寸。这一表示惯例用在器件物理的分析中，但在电子学范围的讨论中仍值得保留。为了避免混淆，把设计值(在版图中用)的符号标为 L' 和 W'，则

$$\begin{aligned} L &= L' - \Delta L \\ W &= W' - \Delta W \end{aligned} \tag{6.19}$$

上式即为电气上考虑的值与设计值之间的关系；ΔL 和 ΔW 是因工艺过程而减小的部分。本

④ 平带电压的名称来源于系统能带图分析，这部分内容已超出目前学习的范围。

章所有公式都将使用电气上考虑的值 W 和 L，并且将在以后的章节中一直保持这个做法。这两个值在 SPICE 中的实际运用将在以后讨论，以使它们最终被解释清楚。

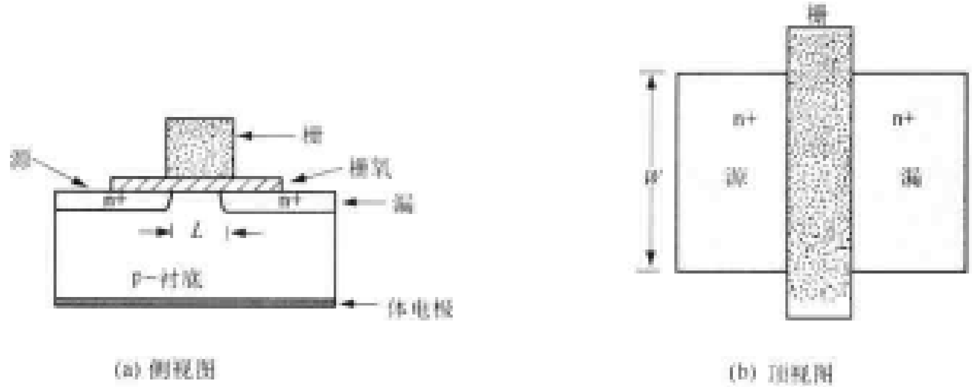


图 6.8　nFET 结构细节

电流特性是通过在物理结构上加上电压，进行物理分析得到的。由图 6.9 可见，电路符号上代表的电压[见图 6.9(a)]和集成结构上外加的电压[见图 6.9(b)]之间有一一对应的关系。为了简化起见，将源端接地，这不会影响结果的一般性，因为只用到相对的电压 V_{GSn} 和 V_{DSn}，现在要做的是确定电流 I_{Dn} 与电压的关系。

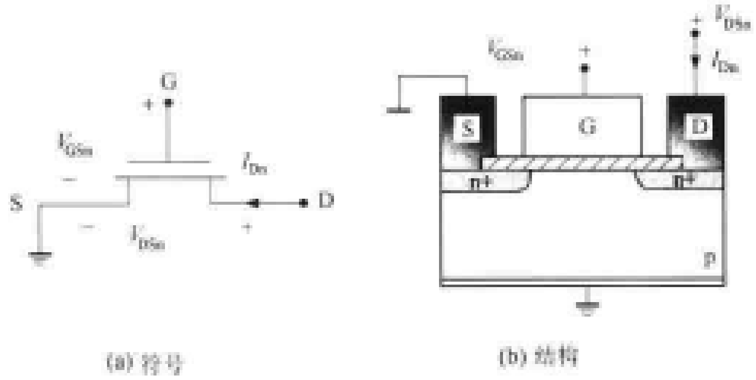
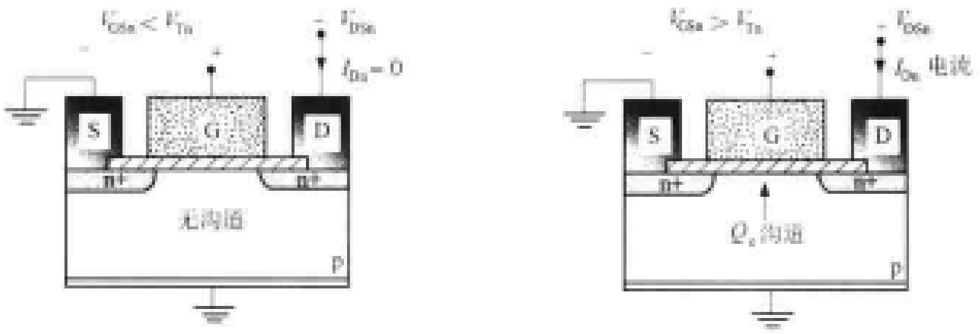


图 6.9　nFET 的电流和电压

了解 n 沟道 MOSFET 中电流情况的关键是，注意 MOS 结构能够通过栅源电压 V_{GSn}，控制栅氧层下面电子电荷层 Q_e 的形成。图 6.10 可以说明这一点。如图 6.10 (a)所示，如果 $V_{GSn} < V_{Tn}$，则 $Q_e = 0$。由于没有电子层存在，所以这两个 n+ 区是相互分隔开的，它们之间没有直接的电流通道存在。从外部来看，漏端与源端之间开路，因此电流 I_{Dn} 必定为 0。这种工作状态称为**截止**，是由 $V_{GSn} < V_{Tn}$，$I_{Dn} = 0$ 定义的。一个截止的晶体管相当于漏端与源端之间一个断开的开关。

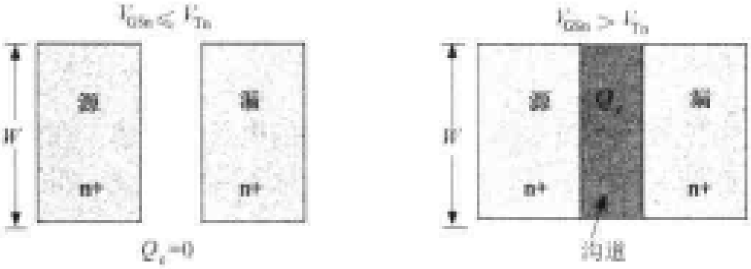
如果栅源电压增加到 $V_{GSn} > V_{Tn}$，情况将发生显著的转变。如图 6.10(b)所示，在栅氧层下面形成了电子电荷层 Q_e。这一电荷层在漏和源 n+ 区的电子之间提供一个电子**沟道**，允许电流在二者之间流动。沟道的存在定义了晶体管的**有源**工作模式。电流 I_{Dn} 的数值同时取决于 V_{GSn} 和 V_{DSn}。



(a) 截止 (b) 导通时偏压

图 6.10 nFET 中沟道的控制

图 6.11 是从工艺层角度来看的 FET 工作模式。这些图表现该器件的工作情况如同它出现在硅片的表面上，而栅层仿佛是透明的。图 6.11(a)是 $V_{GSn} < V_{Tn}$的截止情况；由于 $Q_c = 0$，漏区与源区之间没有沟道存在，器件的作用像一个断开的开关。图 6.11(b)表示的是相反的情况，栅-源电压满足 $V_{GSn} > V_{Tn}$的条件，结果形成了电荷层 Q_c。这定义了有源工作模式。电荷层的作用像两个 n+ 区间的导电沟道，使电荷在二者之间传送。



(a) 截止 (b) 导通状态

图 6.11 nFET 中沟道的形成

上面所描述的行为几乎可以说明，把 nFET 模拟成一个高电平控制开关是合理的，当栅电压小($V_{GSn} < V_{Tn}$)时开关断开，栅电压大($V_{GSn} > V_{Tn}$)时开关闭合。正如第 2 章中所介绍的，在 VLSI 中，开关模型足以用来设计逻辑门。但 FET 的电气特性却与理想开关有很大的偏离。虽然这一点还未影响到用 FET 构成逻辑的规则，但它确实确定了 CMOS 电路瞬态响应基本的上限。鉴于开关速度是现代芯片设计的关键，有必要更深入地研究 MOSFET 的操作，以为 VLSI 设计环境提供一个全面的情况。确定晶体管的尺寸是逻辑门的物理设计和它们的电子操作之间的结合点。

为了在器件层次上了解 nFET 的特性，可以采用把电流画成电压函数曲线的简单方法。由于有两个电压(V_{GSn}和 V_{DSn})，我们将保持其中一个不变而只改变另外一个，并且分别进行两个不同的实验以得到全面的结果。第一个实验见图 6.12，这里将漏-源电压设为电源电压值($V_{DSn} = V_{DD}$)同时将 V_{GSn}的值由 0 V 起正向增加。其结果见 I_{Dn}与 V_{GSn}的关系曲线。当电压 $V_{GSn} < V_{Tn}$时，晶体管截止，$I_{Dn} = 0$。增加栅-源电压至 $V_{GSn} > V_{Tn}$则偏置了 nFET，通过形成电子电荷层 Q_c 使 nFET 进入有源工作区。漏-源电压 V_{DSn}提供移动电荷所需要的电位差，它使电流 I_{Dn}流过该器件。电流值可以用下列公式近似计算

$$I_{Dn} = \frac{\beta_n}{2}(V_{GSn} - V_{Tn})^2 \tag{6.20}$$

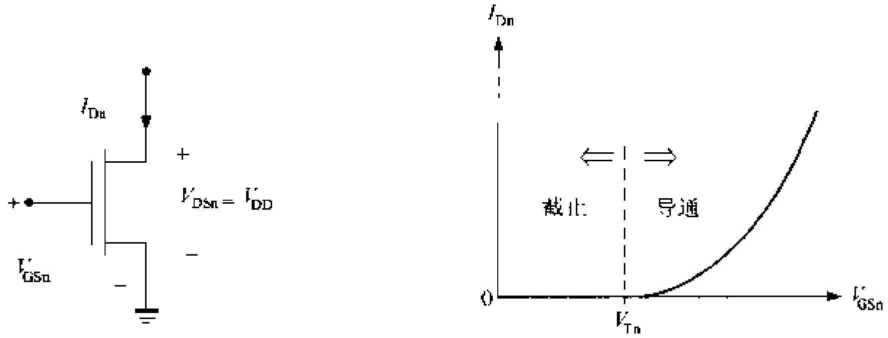


图 6.12 $I-V$ 特性与 V_{GSn} 的关系

上式说明电流与电压的平方相关，它定义了 FET 的**平方律模型**。虽然只是一个近似值，但对计算复杂 CMOS 电路的特性非常有用。与电压相乘的系数 β_n 是器件的互导参数，单位为 A/V^2。每个 nFET 都有自己的 β_n，它由 nFET 的尺寸比决定，即

$$\beta_n = k'_n\left(\frac{W}{L}\right) \tag{6.21}$$

式中，k'_n 是工艺互导参数，由下式计算得到

$$k'_n = \mu_n C_{ox} \tag{6.22}$$

VLSI 设计者不能改变该值。公式中，μ_n 是在硅表面的电子迁移率。在室温下，一个硅 MOSFET μ_n 的典型值在 $(500\sim580)\text{cm}^2/(\text{V}\cdot\text{s})$ 左右，它是材料的特性之一。

注意工艺互导与每单位面积的氧化层电容成正比。

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.23}$$

代入式(6.22)得到

$$k'_n = \frac{\mu_n \varepsilon_{ox}}{t_{ox}} \tag{6.24}$$

所以氧化层薄(t_{ox}值小)，k'_n 值则大，它增加了器件对栅电压的灵敏度，有助于提高器件的开关速度。从物理学的观点来看，减小 t_{ox} 可增大 C_{ox}，从而增强电场效应。

【例 6.1】 考虑一个 nFET，其栅氧层厚度为 $t_{ox}=12$ nm，电子迁移率为 $\mu_n=540\ \text{cm}^2/(\text{V}\cdot\text{s})$。每 cm^2 的氧化层电容为

$$C_{ox} = \frac{(3.9)(8.854\times10^{-14})}{1.2\times10^{-6}} = 2.88\times10^{-7}\ \ \text{F/cm}^2 \tag{6.25}$$

由于介电常数的单位为 F/cm，上式中使 $t_{ox}=12\ \text{nm}=1.2\times10^{-6}$ cm。工艺互导计算为

$$\begin{aligned} k'_n &= \mu_n C_{ox} \\ &= (540)(2.88\times10^{-7}) \\ &= 1.55\times10^{-4}\ \ \text{A/V}^2 \end{aligned} \tag{6.26}$$

或

$$k'_n = 155\ \mu A/V^2 \tag{6.27}$$

如果将氧化层厚度降为 $t_{ox}=8$ nm，则工艺互导增加为

$$k'_n = 233\ \mu A/V^2 \tag{6.28}$$

说明器件更加灵敏。

让我们改变电压至图 6.13 所示的情况。这次，加在 nFET 上的栅-源电压 $V_{GSn} > V_{Tn}$ 不变，改变漏-源电压 V_{DSn}，得到 I_{Dn} 与 V_{DSn} 的关系曲线。当 V_{DSn} 值较小时，电流可由下列公式来估算

$$I_{Dn} = \frac{\beta_n}{2}[2(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2] \tag{6.29}$$

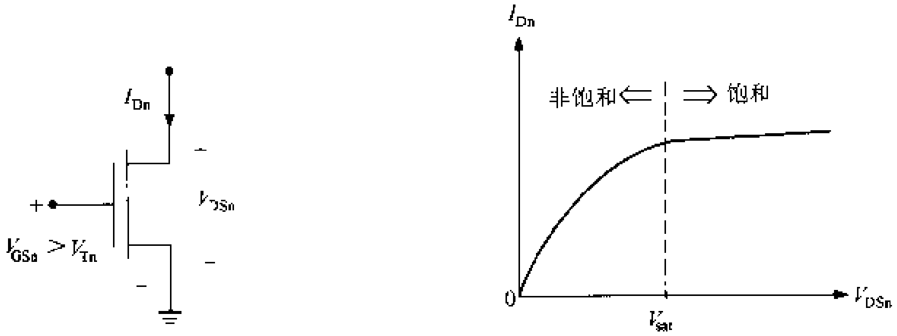


图 6.13　I-V 特性与 V_{DSn} 的关系

这是一条抛物线，峰值点处有

$$\frac{\partial I_{Dn}}{\partial V_{DSn}} = 0 \tag{6.30}$$

求导并令其结果为 0，得到

$$\frac{\partial}{\partial V_{DSn}}[2(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2] = 2(V_{GSn} - V_{Tn}) - 2V_{DSn} = 0 \tag{6.31}$$

该方程的解确定了 V_{DSn} 的一个特定值，称为**饱和电压**

$$\begin{aligned} V_{sat} &= V_{DSn}\big|_{\text{peak current}} \\ &= V_{GSn} - V_{Tn} \end{aligned} \tag{6.32}$$

它表示在图上。对于满足 $V_{DSn} \geq V_{sat}$ 的较大的漏-源电压，电流值几乎与 V_{DSn} 无关，为

$$I_{Dn} = \frac{\beta_n}{2}(V_{GSn} - V_{Tn})^2 \tag{6.33}$$

上式与式(6.20)一致，称为**饱和电流**，因为对于给定的 V_{DSn} 值这是能够流过的电流 I_{Dn} 的最大值。更为详细的分析表明，在 $V_{DSn} \geq V_{sat}$ 时饱和电流略有上升。这一情形常用下式来模拟

$$I_{Dn}=\frac{\beta_n}{2}(V_{GSn}-V_{Tn})^2[1+\lambda(V_{DSn}-V_{sat})] \quad (6.34)$$

式中 λ 是一个经验值,称为**沟道长度调制参数**,单位为 V^{-1}。在手工计算数字电路时,为简单起见,通常假设 $\lambda=0$;若需要可以很容易地把 λ 值的影响考虑到电路的计算机模拟中去。一般可以说,如果 $V_{DSn}\leqslant V_{sat}$,MOSFET 工作在**非饱和区**;如果 $V_{DSn}\geqslant V_{sat}$,则工作在**饱和导电区**。

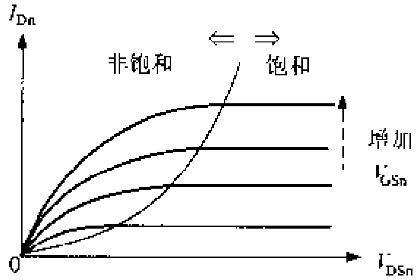


图 6.14 nFET 特性曲线族

图 6.13 中 $I-V$ 曲线显示的电流仅针对一个 V_{GSn} 值。将几个针对不同栅-源电压值的电流曲线叠加在一起就形成了图 6.14 中的**曲线族**。每条线代表一个给定的 V_{GSn} 值。对于一个给定的漏-源电压 V_{DSn},电流随 V_{GSn} 值的增加而增加。非饱和区和饱和工作区的分界线可以由以下饱和电流公式得到

$$I_{Dn}=\frac{\beta_n}{2}V_{sat}^2 \quad (6.35)$$

式中 $V_{sat}=(V_{GSn}-V_{Tn})$ 取决于栅-源电压值。利用这组方程,只要知道电压,就可以求出漏电流 I_{Dn} 的值。

【例 6.2】 考虑一个具有下列特性的 n 沟道 MOSFET:

$$t_{ox}=10\ \text{nm},\ \mu_n=520\ \text{cm}^2/(\text{V}\cdot\text{s}),\ (W/L)=8,\ V_{Tn}=+0.70\ \text{V}$$

利用这些条件,可以求出器件方程。首先利用下式求氧化层电容

$$C_{ox}=\frac{\varepsilon_{ox}}{t_{ox}}=\frac{(3.9)(8.854\times10^{-14})}{10\times10^{-7}}=3.45\times10^{-7}\text{F/cm}^2 \quad (6.36)$$

工艺互导为

$$k'_n=\mu_nC_{ox}=(520)(3.45\times10^{-7})=1.79\times10^{-4}\ \text{A/V}^2 \quad (6.37)$$

或 $k'_n=179\ \mu\text{A/V}^2$。器件互导可以用下式进行计算

$$\beta'_n=k'_n\left(\frac{W}{L}\right)=179(8)=1.435\ \text{mA/V}^2 \quad (6.38)$$

现在让我们来计算不同电压组合下的漏电流。

假设给 nFET 加上电压 $V_{GSn}=2$ V 和 $V_{DSn}=2$ V。首先确定 nFET 的导电状态,即晶体管是工作在饱和区还是在非饱和区?只要了解了这一点,就可以利用合适的公式进行运算。饱和电压为

$$\begin{aligned}V_{sat}&=V_{GSn}-V_{Tn}\\&=2-0.7\\&=1.3\ \text{V}\end{aligned} \quad (6.39)$$

由于 $V_{DS}=2\ \text{V}>V_{sat}$,nFET 是饱和的,所以

$$\begin{aligned}I_{Dn} &= \frac{\beta_n}{2}(V_{GSn} - V_{Tn})^2 \\ &= \left(\frac{1.435}{2}\right)(2-0.7)^2 \\ &= 1.213\ \text{mA}\end{aligned} \tag{6.40}$$

让我们将漏-源电压降低到 $V_{DSn}=1.2$ V，但保持 $V_{GSn}=2$ V 不变。饱和电压仍然是

$$V_{sat} = V_{GSn} - V_{Tn} = 1.3\ \text{V} \tag{6.41}$$

现在 $V_{DSn}=1.2\ \text{V}<V_{sat}$，也就是说晶体管是非饱和的，于是电流可用下式计算

$$\begin{aligned}I_{Dn} &= \frac{\beta_n}{2}[2(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2] \\ &= \left(\frac{1.435}{2}\right)[2(1.3)(1.2)-(1.2)^2] \\ &= 1.21\ \text{mA}\end{aligned} \tag{6.42}$$

这组运算说明了 MOSFET 的一般电流特性。

6.2.1 SPICE Level 1 方程

沟道长度调制的影响，虽然用上面一组公式进行手工计算十分麻烦，但却能很容易地包含到 SPICE 模拟中去。SPICE Level 1 模型采用的另一组 MOSFET 公式如下，在 $V_{DSn}\leqslant V_{sat}$时 MOSFET 的非饱和电流公式为

$$I_{Dn} = \frac{\beta_n}{2}[2(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2](1+\lambda V_{DSn}) \tag{6.43}$$

这使它与以下的饱和电流公式之间有一个连续的过渡：

$$I_{Dn} = \frac{\beta_n}{2}(V_{GSn} - V_{Tn})^2(1+\lambda V_{DSn}) \tag{6.44}$$

上式在 $V_{DSn}\geqslant V_{sat}$时成立。这与物理分析是不一致的，因为沟道长度调制只发生在饱和器件中。然而它却使电流的分析更为简单。这些式子在模拟 CMOS 设计中相当普遍。但是，沟道长度调制效应并不过度地影响数字电路手工计算的结果，所以没有必要使代数计算复杂化。因此，在这里的手工运算中很少用到这些公式。

6.2.2 体偏置效应

到目前为止，一直没有考虑 p 型衬底的存在。事实上，MOSFET 是一个四端器件，它的衬底是器件的**体**(B)端口。如图 6.15 所示，当一个 nFET 的源端和体端间存在电压 V_{SBn}时，就会有**体偏置效应**发生。体偏置电压 V_{SBn}会加大器件的阈值电压，即

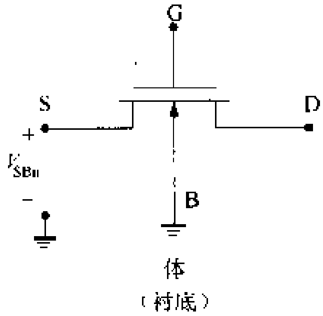


图 6.15　体电极与体偏置电压

$$V_{Tn} = V_{T0n} + \gamma\left(\sqrt{2|\phi_F| + V_{SBn}} - \sqrt{2|\phi_F|}\right) \tag{6.45}$$

式中 γ 是体偏置系数,单位为 $V^{-1/2}$,$2|\phi_F|$ 是公式(6.14)中的体费米电势。V_{T0n} 是体偏置为0时的阈值电压

$$V_{T0n} = V_{Tn}|_{V_{SBn}=0} \tag{6.46}$$

它是工艺规定中的一个值。体偏置系数可用下式估算

$$\gamma = \frac{\sqrt{2q\varepsilon_{Si}N_a}}{C_{ox}} \tag{6.47}$$

式中 $q = 1.6\times10^{-19}$C 为基本电荷单位,$\varepsilon_{Si} = 11.8\varepsilon_0$ 是硅的介电常数,N_a 是 p 型衬底中受主的掺杂数。γ 值通常来自工艺规定。注意,氧化层薄可降低 γ 值。

【例6.3】 有一个 nFET,它的 $V_{T0n} = 0.7$ V,$\gamma = 0.08\ V^{1/2}$,$2|\phi_F| = 0.58$ V。阈值电压取决于体偏置电压 V_{SBn} 的关系为

$$V_{Tn} = 0.70 + 0.08(\sqrt{0.58 + V_{SBn}} - \sqrt{0.58}) \tag{6.48}$$

一些计算结果如下:

图6.16是函数关系的曲线图,它表明一个平方根关系。

V_{SBn}(V)	V_{Tn}(V)
0	0.70
1	0.74
2	0.77
3	0.79

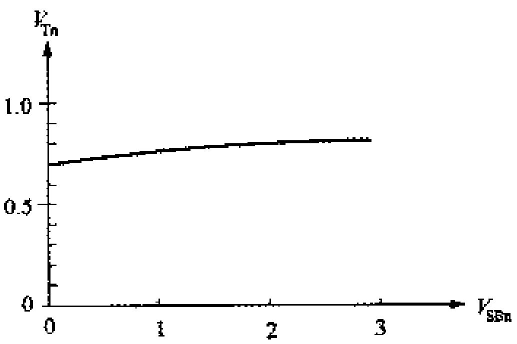


图6.16 体偏置效应

6.2.3 电流方程推导[⑤]

不饱和电流公式是通过分析沟道区的物理过程得来的,即当外加栅源电压 $V_{GSn} > V_{Tn}$ 时在沟道区形成电子电荷密度 Q_e(C/cm²)。图6.17详细描述了这些重要特点。从物理学上讲,加在器件上的漏源电压 V_{DSn} 产生一个电场 E,其方向为从漏指向源(记住,根据定义,漏为高电压一边)。由于电子带有负电荷 $-q$,它们受到一个与电场方向相反的力。因此电子从源端通过沟道流向漏端;这也就是电极名称的来源。在电子学中,通常考虑**惯用电流**,即正电荷移动方向的电流,它与电子流动的方向相反。把这个惯例应用到 nFET 上,电流就是由漏端流向源端的。

⑤ 跳过本节不影响学习的连贯性。读者可从6.3节开始继续讨论主题。

既然进行了物理学方面的定性讨论，现在让我们来做进一步的分析。由电磁学理论知道，电场是守恒场。这意味着，存在有净电位（或电压）$V(y)$，并且

$$E(y) = -\frac{dV}{dy} \tag{6.49}$$

式中，y 是图中所定义的一个坐标，$V(y)$称为**沟道电压**，它是由于外加的漏源电压 V_{DSn}引起的。在沟道两端它的值已知，为

$$V(0) = 0$$
$$V(L) = V_{DSn} \tag{6.50}$$

这两个值是求解方程的边界条件，它们还表明 $V(y)$由漏到源减少。沟道电压的存在改变了沟道中的电荷，使 Q_c 成为坐标 y 的函数。要理解这一点，可以回想一下在简单 MOS 结构（不是一个 FET）中的电荷密度 Q_c 为：

$$Q_c = -C_{ox}(V_{GSn} - V_{Tn}) \quad \text{(MOS 值)} \tag{6.51}$$

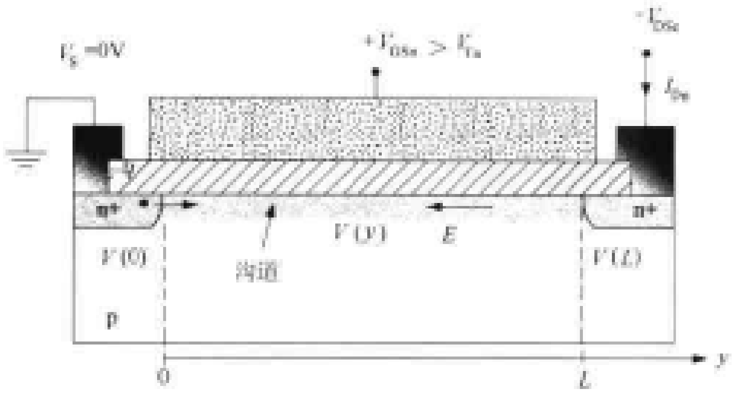


图 6.17　MOSFET 中的沟道电压

式中$(V_{GSn} - V_{Tn})$是加在绝缘氧化层上的有效电压。对于 nFET 来说，由于沟道电压 $V(y)$处在氧化层下面，使情况发生了变化。只要稍微想一下就可以证实由于 $V(y)$是正数，所以它会对抗外加电压 V_{GSn}。因此 nFET 沟道电荷公式为

$$Q_c(y) = -C_{ox}[V_{GSn} - V_{Tn} - V(y)] \quad \text{(MOSFET)} \tag{6.52}$$

该公式表明 Q_c 在沟道中是变化的。最小值在漏一边，为

$$Q_c(L) = -C_{ox}[V_{GSn} - V_{Tn} - V_{DSn}] \tag{6.53}$$

而最大电荷密度则在源端，为

$$Q_c(0) = -C_{ox}[V_{GSn} - V_{Tn}] \tag{6.54}$$

函数关系 $Q_c(y)$是十分有意义的，因为它说明沟道中的电荷密度是不均匀的，这又进一步说明 $I - V$ 的关系是非线性的。

I_{Dn}方程可以通过将上述观察的结果应用到图 6.18 所示的沟道几何图中得到。为了处理变化的电荷密度，让我们从长度为 dy 的沟道微分段着手。电流 I_{Dn}流过该微分段并引起一个电压降为

$$dV = I_{Dn}dR \tag{6.55}$$

式中,dR 是该微分段的微分电阻

$$dR = \frac{dy}{\sigma_n A_n} \tag{6.56}$$

在上式中,σ_n 是电导率,A_n 是断面面积。由于一个 n 型区的电导率为 $\sigma_n = q\mu n$,所以可将分母改写为下列形式

$$\sigma_n A_n = q\mu_n n_e W x_e \tag{6.57}$$

式中 n_e 是单位为 cm^{-3} 的电子密度,x_e 是该点的沟道厚度。沟道电荷密度等于

$$Q_e = -q n_e x_e \tag{6.58}$$

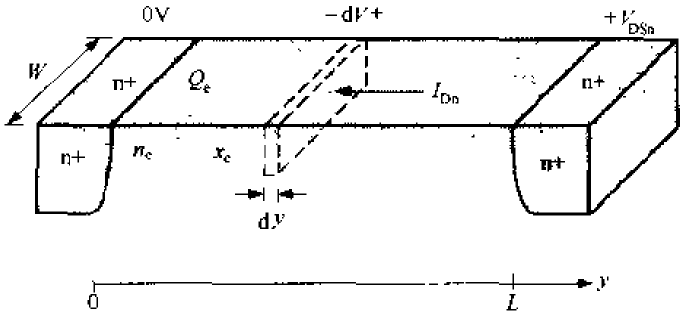


图 6.18 沟道几何尺寸

这一关系很容易看出,只要注意到 Q_e 的单位为 C/cm^2,而式中这些量的组合都是有物理基础的。式中的负号是由于 Q_e 被定义为负值。将它代入电阻方程并利用式(6.52)Q_e 的表达式得到

$$dV = -\frac{I_{Dn} dy}{\mu_n W Q_e} = \frac{I_{Dn} dy}{\mu_n W C_{ox}(V_{GSn} - V_{Tn} - V)} \tag{6.59}$$

整理并积分得到

$$I_{Dn}\int_0^L dy = \mu_n W C_{ox}\int_0^{V_{DSn}}[(V_{GSn} - V_{Tn}) - V]dV \tag{6.60}$$

积分的上下限选择为从 $y=0$ 到 $y=L$,它包括整个沟道。等式右面的电压积分采用这两个点上相应的沟道电压,即 $V(0)=0$ V, $V(L)=V_{DSn}$。假设右边的($V_{GSn} - V_{Tn}$)与沟道电压 V 无关,则

$$I_{Dn}L = \mu_n W C_{ox}[(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2] \tag{6.61}$$

所以

$$I_{Dn} = \mu_n C_{ox}\left(\frac{W}{L}\right)[(V_{GSn} - V_{Tn})V_{DSn} - V_{DSn}^2] \tag{6.62}$$

这与前面方程式(6.29)给出的不饱和电流的表达式一致。

将这一分析扩展到饱和电压 $V_{sat} = (V_{GSn} - V_{Tn})$ 时,我们发现与沟道有关的一个有趣的情况。式(6.53)给出了漏端的沟道电荷,代入饱和电压 $V_{DSn} = V_{sat}$ 得到

$$Q_e(L) = -C_{ox}[V_{GSn} - V_{Tn} - V_{sat}] = 0 \tag{6.63}$$

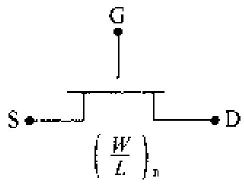
即在饱和电压时电荷浓度下降至0。更详细的分析显示，电荷并未真正下降至0，而是实际上很小。这种现象称为nFET中的**沟道夹断**。正式地说，它是饱和工作区和不饱和工作区的边界。对于 $V_{DSn} > V_{sat}$，电荷的夹断限制了电流（所以称为**饱和**），而夹断效应本身缩短了沟道的有效长度（这就是**沟道长度调制**系数 λ 的来源）。

6.3 FET的RC模型

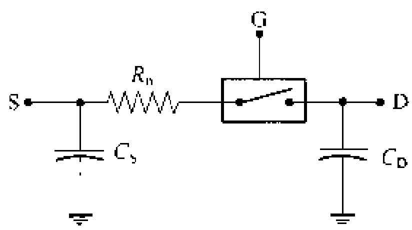
上面的电流方程说明nFET表现出非线性的 $I-V$ 特性。这一特点使我们很难分析采用FET的电路，因为电路方程本身也变成非线性的了，从而使手工运算也变得十分枯燥繁重。自然，解决的办法是使用一个像SPICE这样的CAD工具来进行困难的分析。这并不能解决VLSI设计者所面临的问题：他们必须**构建**一个具有合适电特性的电路。这就突出了**分析**和**设计**之间的区别：分析是研究设计过程得到的新电路，而设计者是真正的问题解决者，他们要用已有的知识作为基础来构建新的系统。

有两种途径可以解决复杂的晶体管方程问题。第1种途径是让电路专家来处理非线性器件所带来的问题。熟练的电子设计者在芯片设计过程中是不可缺少的。另一方面，VLSI系统设计是以逻辑与数字结构为基础的；工作在系统层次上的工程师也必须懂得FET电路。这也是第2种途径的基础：建立起器件的在逻辑和系统层次上有用的简**化线性模型**。这个模型就它的本质而言将忽略有关电流的大部分细节，却使它在系统层次上分析复杂电路的信号流简单得多。如果能在模型中包括晶体管的至少一部分重要特性，那么它就可以为设计的第1阶段提供基础。简化的线性模型也使我们可以开发出能比较各种算法的技术，以选择最有效的VLSI方法。

在处理问题时将采用的线性模型见图6.19。它将nFET简化为一个电阻 R_n，两个电容器（C_S 和 C_D），以及一个高电平逻辑控制开关。各线性元件的值取决于nFET的宽长比 $(W/L)_n$，取决的方式将在后面两小节中讨论。



(a) nFET符号



(b) nFET的线性模型

图6.19 nFET的RC模型

6.3.1 漏源FET电阻

场效应晶体管本质上是非线性的，所以对用固定 R_n 值的线性电阻来模拟通过nFET的电流这一想法需要非常小心。

考虑图6.20中的情况。在图6.20(a)中，假定栅源电压被设定在一个值 $V_{GSn} > V_{Tn}$，使

nFET 成为有源器件。电流 I_{Dn}于是成为漏源电压的函数,如图 6.20(b) 所画的曲线。曲线上任意一点的漏源电阻为

$$R_n = \frac{V_{DSn}}{I_{Dn}} \tag{6.64}$$

非线性效应是由于 I_{Dn}随 V_{DSn}而变化的,因此使 R_n 本身成为 V_{DSn}的函数。

这种相关性的影响可以通过图中 a,b,c 三点电阻的公式来发现。当 V_{DSn}较小时(a 点),忽略非线性电流公式(6.29)中的平方项 V_{DSn}^2,此时电流可近似为

$$I_{Dn} \approx \beta_n(V_{GSn} - V_{Tn})V_{DSn} \tag{6.65}$$

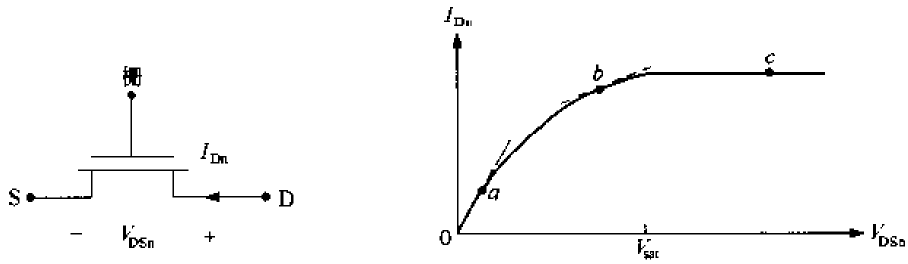


图 6.20 确定 nFET 电阻

于是电阻为

$$R_n \approx \frac{1}{\beta_n(V_{GSn} - V_{Tn})} \tag{6.66}$$

所以 R_n 随 V_{GSn}而变化。在点 b 必须使用完全不饱和电流公式,所以

$$R_n = \frac{2}{\beta_n[2(V_{GSn} - V_{Tn}) - V_{DSn}]} \tag{6.67}$$

上式说明 R_n 同时是 V_{GSn}和 V_{DSn}二者的函数。当器件在 c 点达到饱和时,利用忽略沟道长度调制公式(6.20),电阻成为

$$R_n = \frac{2V_{DSn}}{\beta_n(V_{GSn} - V_{Tn})^2} \tag{6.68}$$

电阻再次同时随 V_{GSn}和 V_{DSn}而变化。

这些公式说明,不可能将 R_n 定义为一个常数值而同时仍然保持电流有上述正确的行为。但是注意,在所有情况下,R_n 都与 β_n 成反比,即

$$R_n \propto \frac{1}{\beta_n} \tag{6.69}$$

简单地说就是,一个 β_n 大的器件较之一个 β_n 小的器件能够导通更多的电流。由定义

$$\beta_n = k'_n\left(\frac{W}{L}\right)_n \tag{6.70}$$

可见器件的宽长比$(W/L)_n$ 是重要的参数。定性地说,nFET 宽度 W 的增加会降低它的电阻。

清楚了这一点，我们将引入一个简单公式，把电阻模拟成与晶体管宽长比(或宽度)有关的函数，即

$$R_n = \frac{\eta}{\beta_n(V_{DD} - V_{Tn})} \tag{6.71}$$

在建立该公式时，比照上面的表达式将电源电压 V_{DD}作为 V_{GSn}最大可能的值。式中引入因子 η 是考虑当晶体管切换通过不同工作区时所发生的一些变化；它没有什么物理依据。在文献中，这个乘数因子的范围一般在 $\eta=1$ 到 $\eta=6$ 左右。为了简单起见，将选择 $\eta=1$，肯定由此得到的数值会小些。公式于是简化为

$$R_n = \frac{1}{\beta_n(V_{DD} - V_{Tn})}\ \Omega \tag{6.72}$$

这是最终的形式。电阻 R_n 的单位是欧姆，这与式中由分母所确定的单位是一致的。

【例 6.4】 有一 nFET，其沟道宽 $W=8\ \mu m$，沟道长 $L=0.5\ \mu m$，采用的工艺使 $k'_n=180\ \mu A/V^2$，$V_{Tn}=0.70$ V，$V_{DD}=3.3$ V。线性化的漏源电阻计算为

$$R_n = \frac{1}{\beta_n(V_{DD} - V_{Tn})} \tag{6.73}$$

代入值得到

$$R_n = \frac{1}{(180\times10^{-6})\left(\frac{8}{0.5}\right)(3.3-0.7)} = 133.5\ \Omega \tag{6.74}$$

若将沟道宽度缩小至 $W=5\ \mu m$，而保持所有其他量不变，则电阻增加到

$$R_n = 133.5\left(\frac{8}{5}\right) = 213.6\ \Omega \tag{6.75}$$

这里我们只是根据 R_n 与沟道宽度成反比的关系把电阻值放大。应当记住，这些值并非 nFET 电阻的实际值，而只是用在简化的模型中。

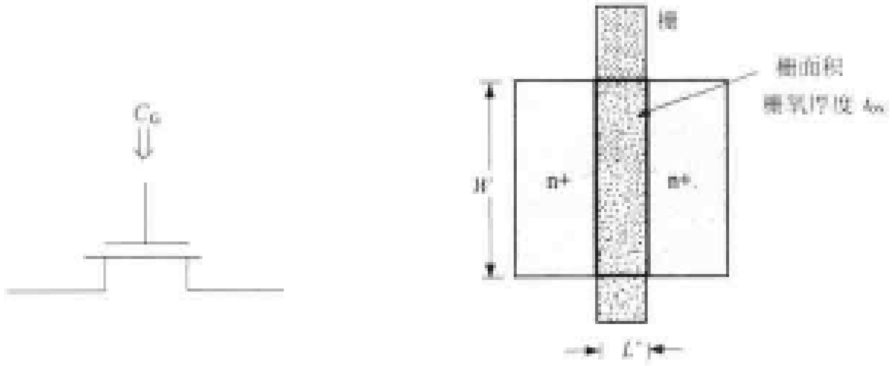
6.3.2 FET 电容

一个 MOSFET 有几个寄生电容，它们必须包括在简化的开关模型中。正如以后将会看到的，CMOS 电路的最大开关速度是由电容决定的。

1. MOS 电容

金属氧化物半导体的工艺层次本质上是一个电容器，所以让我们首先来分析一下它的值。电路模型见图 6.21(a)。如果从 FET 的栅端看进去，可以看到由 MOS 结构形成的**栅电容** C_G。由于这个区域有一个厚度为 t_{ox}的栅氧层，所以可用单位面积的氧化层电容 C_{ox}来描述。当用 A_G 表示栅区面积时

$$C_G = C_{ox}A_G \tag{6.76}$$



(a) 电路图　　(b) 实际来源

图 6.21　FET 的栅电容

单位是法拉，这是栅端和地之间的电容。对于图 6.21(b)中简单的几何图形，栅面积为 $A_G = WL'$，其中 W 为沟道宽度，L'为设计的沟道的长度。L'就是版图顶视图上由栅区范围定义的沟道长度。因此

$$C_G = C_{ox}WL' \tag{6.77}$$

上式得出一个重要的结果，即栅电容与沟道宽成正比。

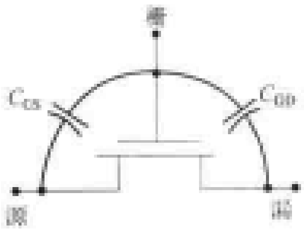


图 6.22　栅源和栅漏电容

也可以用栅源电容 C_{GS}和栅漏电容 C_{GD}来描述 MOS 的另外两个寄生电容，如图 6.22 所示。这两个寄生电容是很复杂的，它们的值随沟道区形状改变引起的电压改变而改变。当 $C = C(V)$时，电容就称为**非线性**的。在 VLSI 系统设计中，通常利用像 SPICE 这样的电路模拟程序来进行细节计算。就现在的目的，将用下式简单估算电容值

$$C_{GS} \approx \frac{1}{2}C_G \approx C_{GD} \tag{6.78}$$

换句话说，我们只是把栅电容除以 2，将它平分为 C_{GS}和 C_{GD}。虽然不很精确，但这样就能使我们把注意力集中在大的特性上。适当利用一组 CAD 工具就能得到最后的验证。

【例 6.5】　有一 FET，其氧化层电容为 $C_{ox} = 3.45 \times 10^{-7}\,\text{F/cm}^2$，栅的尺寸为 $W = 8\,\mu\text{m}$，$L' = 0.5\,\mu\text{m}$。由栅电容公式得到

$$C_G = (3.45\times10^{-7})(8\times10^{-4})(0.5\times10^{-4}) \tag{6.79}$$

虽则这是一个简单的计算，倘若注意到 $C_{ox} = 3.54 \times 10^{-7} = 3.45\ \text{fF}/\mu\text{m}^2$，而 $1\ \text{fF} = 10^{-15}\text{F}$，则计算还可进一步简化，即

$$C_G = 3.45(8)(0.5) = 13.8\ \text{fF} \tag{6.80}$$

于是，栅源和漏源的电容可估计为

$$C_{GS} \approx \frac{1}{2}C_G = 6.9 \text{ fF} = C_{GS} \tag{6.81}$$

这些是 FET 电容值的典型数量级。应当记住，我们总是与仅为几个 fF 数量级的器件电容打交道。

2. 结电容

半导体物理告诉我们，一个 pn 结自身就表现出电容，这是由于存在相反极性的电荷。这个电容称为**结**电容或**耗尽**电容，存在于 FET 的每个漏区和源区。图 6.23 表示存在 pn 结和相关的电容 C_{SB}（源-体）与 C_{DB}（漏-体）。我们通常引入一个单位为 F/cm^2 的参数 C_j 来描述该电容的特性，于是总电容为

$$C_0 = C_j A_{pn} \text{ F} \tag{6.82}$$

其中 A_{pn}是结面积，单位为 cm^2。C_j 值取决于工艺过程，且随掺杂程度而改变。

在将此公式用于 nFET 时有两个比较复杂的问题。第 1 是该电容也随电压而改变。当加上一个反偏压 V_R 时，通常可以用下式模拟为

$$C = \frac{C_0}{\left(1 + \frac{V_R}{\phi_0}\right)^{m_j}} \tag{6.83}$$

这里 C_0 是**零偏电容**（此时 $V_R = 0$），ϕ_0 是结的**内建电势**，而 m_j 称为结的**台阶系数**。ϕ_0 和 m_j 都由掺杂特性所决定。有一种特殊情况是**突变或阶梯结**，在此处掺杂由常数受主浓度 N_a 突变为常数施主浓度 N_d。在这种情形中 $m_j = 1/2$，同时内建电压可计算为

$$\phi_0 = \left(\frac{kT}{q}\right) \ln\left[\frac{N_d N_a}{n_i^2}\right] \tag{6.84}$$

另一个简单模型是**线性渐变结**，它的掺杂浓度的变化是位置的线性函数。此时的台阶系数 $m_j = 1/3$，如果已知掺杂细节就可以计算出内建势 ϕ_0。就我们目前的目的，总是假设 C_j，ϕ_0 和 m_j 为已知参数。一般来说，电容的最大值为 $C = C_0$，此时 $V_R = 0$；增加结上的反相电压会使 C 值下降，如图 6.24 所示。我们在手工计算中将用零偏的值作为估计值，而在需要更精确值时再使用 CAD 工具。

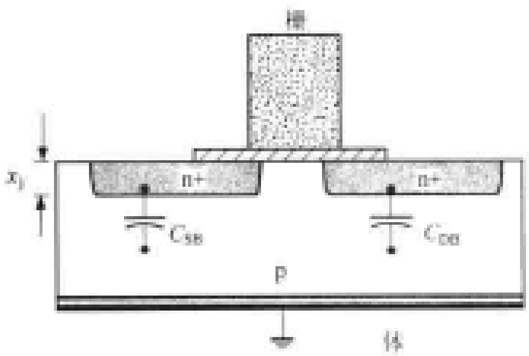


图 6.23　MOSFET 中的 pn 结电容

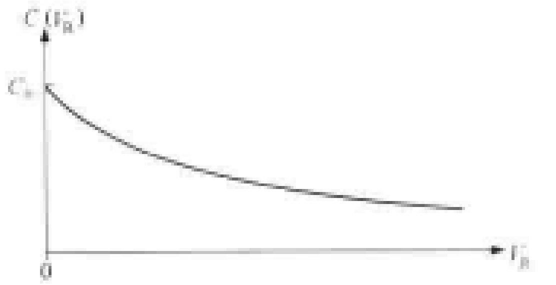


图 6.24　结电容随反相电压变化

在计算 pn 结电容时需要考虑的第 1 个复杂问题是 pn 结的几何形状。如图 6.23 断面图所示，n+ 区被"嵌埋"在 p 型衬底中的深度为 x_j(称为结深)。在计算 pn 结的面积 A_{pn}时，必须注意把底和侧边的影响都包括进去。图 6.25 表示的是几何形状。图 6.25(a)FET 的顶视图确定了晶体管的沟道宽度 W 和 n+ 区的范围 X(由栅向外)。计算 pn 结面积的三维尺寸见图 6.25(b)。由于 n+ 区可以看成一个"没有盖的盒子"结构，所以可将边界分解为底部和四壁。很容易看到底部面积为

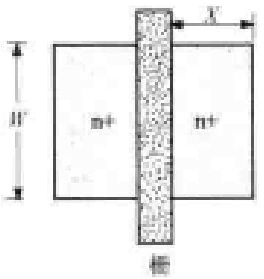
$$A_{bot} = XW \tag{6.85}$$

它等于顶视图上看到的 n+ 区面积。当将该区每单位面积的零偏结电容用符号 C_j 表示，单位为 F/cm² 时，底部部分的电容为

$$C_{bot} = C_j XW \tag{6.86}$$

为了计算侧壁电容 C_{sw}，注意到侧壁的总面积为四壁之和。各个侧壁部分的高等于结深 x_j。侧壁 1，侧壁 2 的面积为($W \times x_j$)，而侧壁 3 和侧壁 4 的面积为($X \times x_j$)。将各项相加得到

$$\begin{aligned} A_{sw} &= 2(W \times x_j) + 2(X \times x_j) \\ &= x_j P_{sw} \end{aligned} \tag{6.87}$$



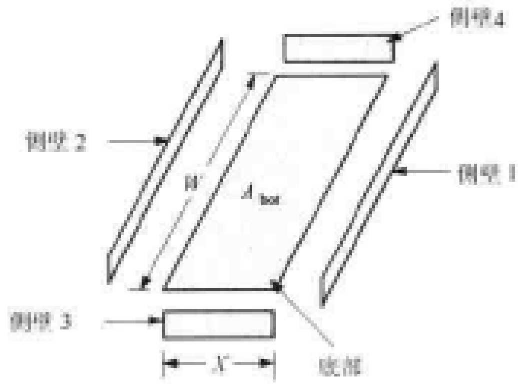


图 6.25 计算 FET 的结电容

式中 P_{sw}是周长，单位为 cm。这样对于本例中的长方形几何图形

$$P_{sw} = 2(W + X) \tag{6.88}$$

将其乘以单位面积的结电容就得到侧壁电容，通常将它修改成下列形式

$$C_{sw} = C_{jsw} P_{sw} \quad \text{F} \tag{6.89}$$

式中

$$C_{jsw} = C_j x_j \ \text{F/cm} \tag{6.90}$$

为**单位周长侧壁电容**。由于周长 P_{sw}可直接从版图中找到，所以使用该式非常方便。在实际中，C_{jsw}规定为一个工艺参数，而 C_j 本身就是指底部电容。

这些公式都忽略了栅下 n+ 区与栅的重叠部分 L_o。在手工计算中，应在所有的地方将 X 替换成

$$X \to (X + L_o) \tag{6.91}$$

来包括这一重叠部分。在 SPICE 模拟中，采用设计尺寸 L' 和 W'（原文为 L 和 W ——译者注）来描述电路，同时将栅重叠（和对其他方面）的修正也包含在模拟信息中。

n+ 区的总零偏电容是将底部和侧壁的电容加在一起得到的

$$\begin{aligned} C_n &= C_{bot} + C_{sw} \\ &= C_j A_{bot} + C_{jsw} P_{sw} \end{aligned} \tag{6.92}$$

用这些可以计算出 C_{SB} 和 C_{DB}。值得注意的是，底部与侧壁结的非线性特性通常是不同的，所以可以得到包括这两个不同非线性形式的公式

$$C_n = \frac{C_j A_{bot}}{\left(1+\frac{V}{\phi_0}\right)^{m_j}} + \frac{C_{jsw} P_{sw}}{\left(1+\frac{V}{\phi_{osw}}\right)^{m_{jsw}}} \tag{6.93}$$

式中 V 是反向电压，m_j 和 ϕ_o 描述底部结，m_{jsw} 和 ϕ_{osw} 则是侧壁参数。这些都作为常规包含在 SPICE 模拟中。

6.3.3 模型建立

现在可把寄生电阻和电容的影响合在一起，建立 nFET 的简单 RC 模型。画出一个版图布局有助于理解这个模型。图 6.26 是一个包含各部分电容的 nFET 顶视图。晶体管周围的 p 型衬底处于接地电位。从任意一边进入的信号同时看到 MOS 电容（C_{GS} 或 C_{GD}）和结寄生电容（C_{SB} 或 C_{DB}）。

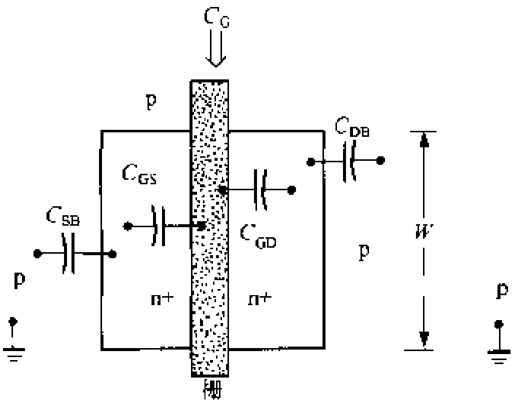


图 6.26　FET 电容直观图

物理版图是图 6.27(a) 中电路图的基础，图中的电容器被分成源端和漏端部分。最简单的方法是将其写成

$$\begin{aligned} C_S &= C_{GS} + C_{SB} \\ C_D &= C_{GD} + C_{DB} \end{aligned} \tag{6.94}$$

上式把与给定节点连在一起的所有电容加在一起来近似总的电容。为了简单起见，将用零偏的电容值进行所有的手工运算。应当注意的是，电阻 R_n 与宽长比 (W/L) 成反比，而电容则随沟道的宽度 W 而增加。

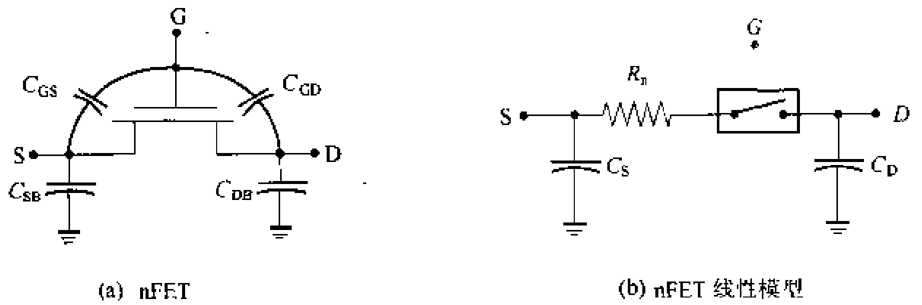
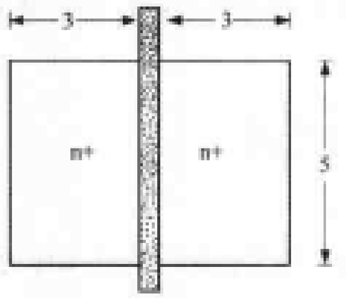


图 6.27　nFET RC 模型的最终结构

【例 6.6】 让我们为图 6.28 中的 nFET 建立一个开关模型，度量单位用微米(μm)表示。由于已知重叠的距离 $L_o=0.05$ μm，导电沟道长 $L=0.5-2(0.05)=0.4$ μm，沟道宽如图为 $W=5$ μm。假设电源电压为 $V_{DD}=3.3$ V，则寄生电阻为

$$R_n = \frac{1}{\left(\frac{5}{0.4}\right)(150\times10^{-6})(3.3-0.6)} = 197.5\ \ \Omega \tag{6.95}$$



$L'=0.5$ μm　　$k'_n=150$ μA/V²
$L_o=0.05$ μm　　$C_{ox}=2.70$ fF/μm²
$V_{Tn}=0.6$ V　　$C_j=0.86$ fF/μm²
$C_{jsw}=0.24$ fF/μm

图 6.28　用于建模例子的几何图

如果 n+ 区的薄层电阻已知，就能求出寄生电阻，并加到该值上。

栅电容为

$$C_G = (2.7)(5)(0.5) = 6.75\ \ \text{fF} \tag{6.96}$$

所以取栅值的一半得到

$$C_{GS} = C_{GD} = 3.375\ \ \text{fF} \tag{6.97}$$

源端或漏端的结电容为

$$C_n = (0.86)A_{bot} + (0.24)P_{sw} \tag{6.98}$$

由于重叠部分 $L_o=0.05$ μm，面积和周边都分别比设计值(3×5)μm 和 16 μm 要大。将这一因素考虑到公式中得到

$$\begin{aligned} C_n &= (0.86)(5)(3.05)+(0.24)(2)(5+3.05) \\ &= 16.98\ \ \text{fF} \end{aligned} \tag{6.99}$$

于是最终的漏和源电容为

$$C_D = C_S = 16.98+3.375 = 20.36\ \ \text{fF} \tag{6.100}$$

计算完毕。

这一简单模型为设计估算提供了一个合理的基础。为了将其用于解决电路问题，只要用这个模型代替晶体管，然后运用标准的线性电路技术。由于它忽略了 FET 内在的非线性特点，其分析的精度是有限的。在初始设计产生一个备选电路之后，由计算机模拟可以提高精度。简化的器件建模是 VLSI 设计过程的一个重要部分，因为它使我们能够很快地建立起一个基本电路。这些电路总是要用 CAD 工具来检查且做精细的调整。

6.4 pFET 特性

一个 p 沟道的 MOSFET 与一个 nFET 在电气上是互补的。由第 2 章看到 nFET 模型像一个高电平控制开关,而 pFET 的行为则像一个低电平控制开关。在器件层次上,它们的互补特性甚至更为明显。假定从一个 nFET 开始,并希望将它修改成一个 pFET,我们对这一结构所要做的全部就是

- 将所有的 n 型区改为 p 型区
- 将所有的 p 型区改为 n 型区

这样得到的器件事实上就是一个 pFET,见图 6.29。我们在两个器件中都选择了 p 型衬底,所以需要一个 n 阱区来嵌入 pFET。两个器件都假设具有相同的氧化层厚度 t_{ox},所以

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.101}$$

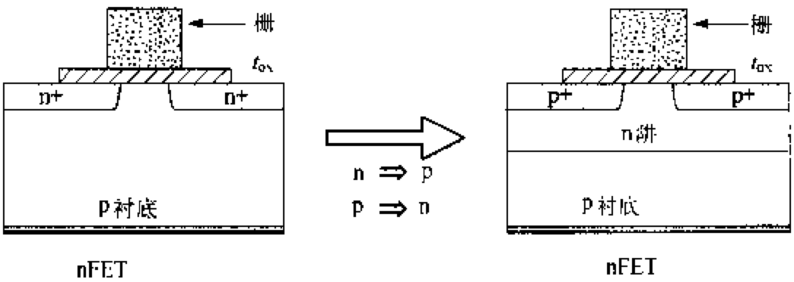
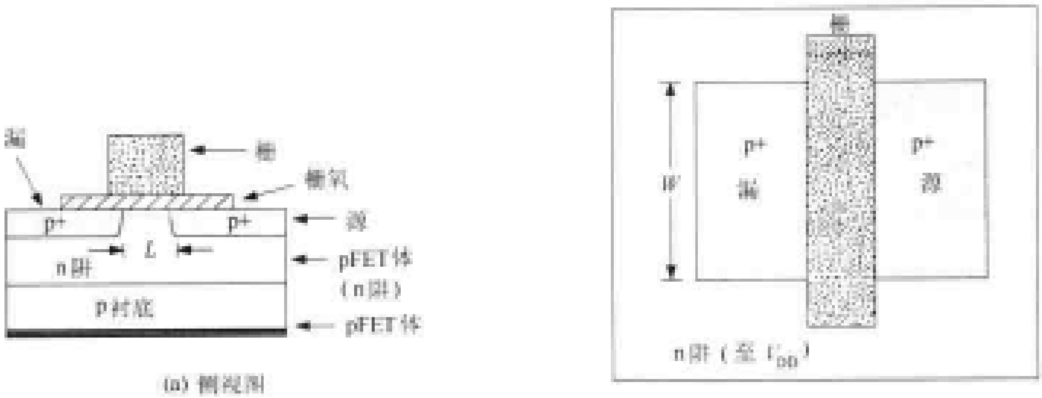


图 6.29 修改 nFET 为 pFET

同时描述了 nFET 和 pFET。这意味着场效应的基本机理与我们对 nFET 的讨论是一致的。不过,既然这两个区域的极性相反,它们的电场方向和电荷极性也是相反的。

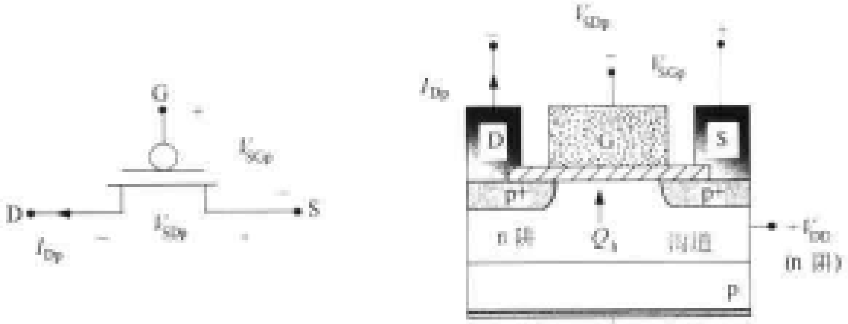
图 6.30 是一个 pFET 的结构细节。如图 6.30(a)的侧视图所示,沟道长度 L 定义为源与漏 p+ 区边之间的距离,而沟道宽则如图 6.30(b)顶视图所示由 p+ 区的范围确定;这些特征尺寸与用来定义 nFET 的尺寸是一样的。左右两个图中都显示了 n 阱的存在,它是 pFET 的重要区域,因为它的作用是作为该器件的体电极。从电学上来讲,n 阱与正电源电压 V_{DD}相连,它的作用是保证电压能很好地确定。对于 nFET,源和漏的命名要求我们了解它们的相对电平。不过 pFET 确定源漏的定义与 nFET 的正好相反。这意味着,具有较高电压的 p+ 一边为源,而另一边(处于低电压)为漏。

p 沟道 MOSFET 利用带正电的空穴传导电流。图 6.31 定义了 pFET 的电流 I_{Dp}和器件电压,左右两个图都假设器件的右边是源。首先注意,图 6.31(a)中的电路符号表明电流 I_{Dp} 由漏电极流出。这是因为正电荷从源向漏移动,它决定了电流的方向。pFET 的电压参照源端且用符号 V_{SGp}(源-栅电压)和 V_{SDp}(源-漏电压)表示;注意,它们与 nFET 相类似的量 V_{GSn} 和 V_{DSn}的极性相反。图 6.31(b)的结构图也表明 n 阱层在电气上是与电源电压 V_{DD}相连的。



(a) 侧视图　(b) 顶视图

图 6.30　pFET 的结构细节



(a) 符号　(b) 结构

图 6.31　pFET 中的电流和电压

通过 pFET 导电是由源栅电压 V_{GSp}控制的。由栅、氧化层和 n 阱各层构成的 MOS 结构的特性体现在 pFET 的阈值电压 V_{Tp}上。按一般惯例，V_{Tp}是一个负数，典型值为 $V_{Tp}=-0.5$ V到 $V_{Tp}=-1.0$ V 左右。从物理学的观点来看，V_{SGp}值的大小决定了栅电压相对于源电压是否足够负以能在栅氧层下面形成一个空穴层，从而建立起一个正的空穴电荷密度 Q_h (C/cm^2)以在源漏之间形成一个沟道。这可以总结为

$$\begin{aligned}&\text{当 } V_{SGp}<|V_{Tp}|\text{ 时，}Q_h=0\\&\text{当 } V_{SGp}>|V_{Tn}|\text{）时，}Q_h\text{ 存在}\end{aligned}\qquad(6.102)$$

这里，采用了阈值电压的绝对值$|V_{Tp}|$。第 1 行对应的情况是栅电压没有足够地负，不能够在 n 阱形成一个空穴导电层。而第 2 种情况是 V_{SGp}足够大，可以保证栅电压吸引空穴并且形成沟道。源漏电压 V_{SDp}的作用是如果沟道存在就把电荷从源移向漏端。

pFET 的阈值电压可用下式计算

$$V_{Tp}=-\frac{1}{C_{ox}}\sqrt{2q\varepsilon_{Si}N_d(2\phi_{Fp})}\ -2\phi_{Fp}+V_{FBp}\mp\frac{qD_I}{C_{ox}}\qquad(6.103)$$

式中 N_d 是 n 阱中施主的掺杂密度，

$$2\phi_{Fp}=2\left(\frac{kT}{q}\right)\ln\left(\frac{N_d}{n_i}\right)\qquad(6.104)$$

它是在 pFET 中形成空穴层所需要的表面电位，V_{FBp}是 pFET MOS 结构的平带电压，而最后一项代表调节阈值的离子注入剂量 D_I。负号'−'用在注入施主时，而正号'+'则对应于注入受主的情况。

图 6.32 概括了一个 pFET 的导电模型。图 6.32(a)是截止的情况。此时，$V_{SGp}<|V_{Tp}|$，所以 $Q_h=0$，没有任何沟道存在。这使 $I_{Dp}=0$，所以可以模拟为一个断开的开关。图 6.32(b)表示的是由 $V_{SGp}\geqslant|V_{Tp}|$定义的导通状态。此时空穴导电层形成并且构成沟道。由于电场方向是从右指向左，带正电荷的空穴由源(右)端出发流向漏(左)端，因而如图那样，pFET 电流 I_{Dp}流出漏电极。

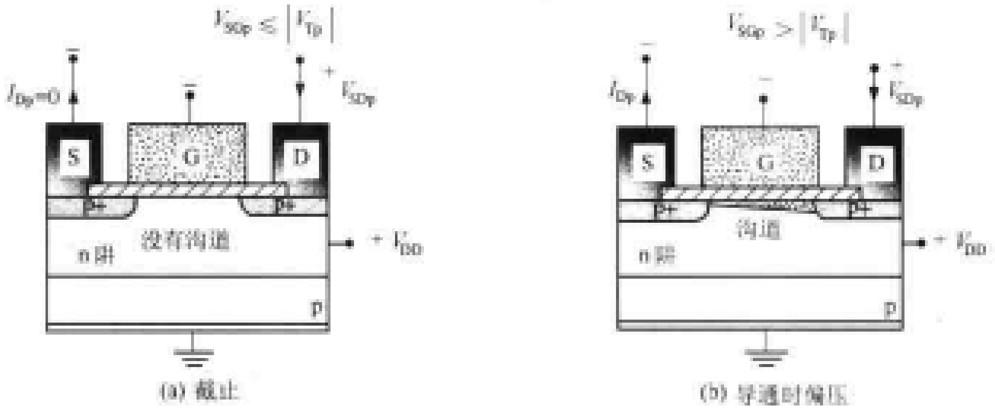


图 6.32　pFET 的导电模型

pFET 的电流-电压特性可用介绍 nFET 相同的方法来描述。如图 6.33 所示，源漏电压 V_{SDp}被设定为 V_{DD}(电源电压值)，同时增加源栅电压 V_{SGp}。对于 $V_{SGp}\leqslant|V_{Tp}|$，由于没有沟道存在，器件截止，$I_{Dp}=0$。当 V_{SGp}上升到$|V_{Tp}|$以上时，电荷层 Q_h 形成，器件导通。电流可以用平方定律公式来近似：

$$I_{Dp}=\frac{\beta_p}{2}(V_{SGp}-|V_{Tp}|)^2 \tag{6.105}$$

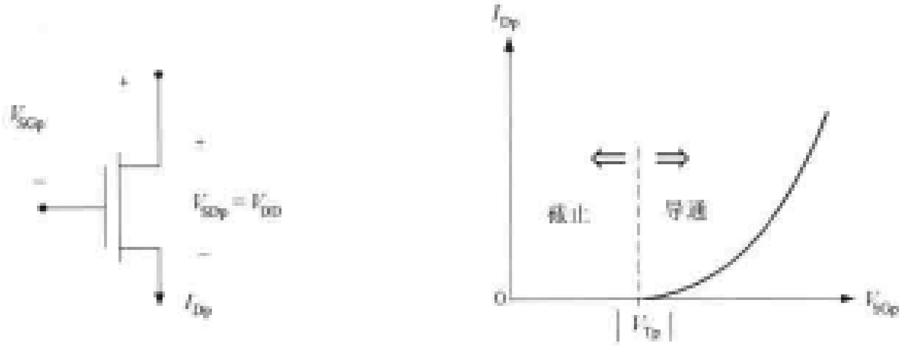


图 6.33　栅控制的 pFET 电流-电压特性

式中

$$\beta_p=k'_p\left(\frac{W}{L}\right)_p \tag{6.106}$$

是 pFET 器件的互导,单位为 A/V^2。$(W/L)_p$ 是 pFET 的宽长比,k'_p 是 pFET 的工艺互导参数

$$k'_p = \mu_p C_{ox} \tag{6.107}$$

单位为 A/V^2。在该公式中,μ_p 是空穴迁移率。这些定义与 nFET 的参数一样,只是必须用 μ_p 来描述空穴在硅中的运动。室温下硅表面空穴迁移率的典型值是 $\mu_p = 220\ cm^2/(V\cdot s)$;这显著低于前面提到的电子迁移率(约为 $550\ cm^2/(V\cdot s)$)。它们的典型比率为

$$r = \frac{\mu_n}{\mu_p} \approx 2 \sim 3 \tag{6.108}$$

注意,FET 电流中的重要乘法因子是互导系数

$$\beta_n = k'_n\left(\frac{W}{L}\right)_n$$
$$\beta_p = k'_p\left(\frac{W}{L}\right)_p \tag{6.109}$$

当 nFET 和 pFET 用于同一电路时,k'_p 与 k'_n 间的差别使 $(W/L)_n$ 和 $(W/L)_p$ 的设计选择会有某些特点。

图 6.34 表示的是较为一般的情形,即 V_{SGp} 保持不变而 V_{SDp} 增加。从 V_{SGp} 的每一个值可以得到一条不同的 I_{Dp} 对应 V_{SDp} 的曲线,从而形成一个曲线族。对于一个 pFET,其饱和电压由下式确定

$$V_{sat} = V_{SGp} - |V_{Tp}| \tag{6.110}$$

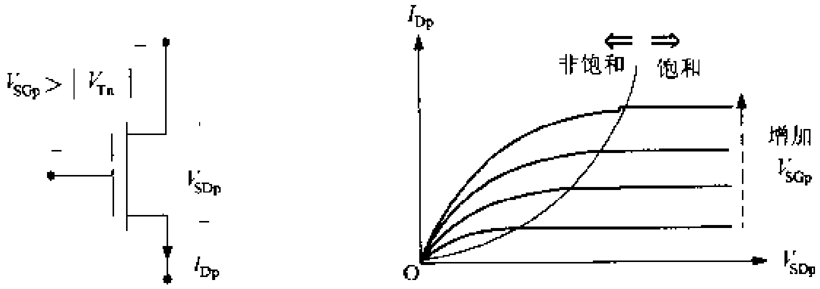


图 6.34 pFET I-V 特性曲线族

非饱和导电发生在 $V_{SGp} \leq V_{sat}$ 时,并可描述为

$$I_{Dp} = \frac{\beta_p}{2}[2(V_{SGp} - |V_{Tp}|)V_{SDp} - V_{SDp}^2] \tag{6.111}$$

而饱和则发生在 $V_{SDp} \geq V_{sat}$ 时,并有

$$I_{Dp} = \frac{\beta_p}{2}(V_{SGp} - |V_{Tp}|)^2 \tag{6.112}$$

饱和导电已画在前面图 6.33 中;如果一个 FET 源漏之间的电压较大(与 V_{sat} 相比),则可以判别出它处于饱和状态。

pFET 寄生参数

pFET 的寄生电阻和电容的计算方式与 nFET 相同。一个线性化的 pFET 电阻为

$$R_p = \frac{1}{\beta_p(V_{DD} - |V_{Tp}|)} \tag{6.113}$$

它表明了下述相关性

$$R_p \propto \frac{1}{\beta_p} = \frac{1}{k'_p(W/L)_p} \tag{6.114}$$

宽长比大则电阻小,允许通过的电流大。

电容的计算也用与 nFET 同样的公式。例如输入栅电容为

$$C_{Gp} = C_{ox}(WL)_p \tag{6.115}$$

式中 C_{ox}对于这两种晶体管是一样的。栅源和栅漏电容可近似为

$$C_{GS} \approx \frac{1}{2}C_{Gp} \approx C_{GD} \tag{6.116}$$

p+ /n 结的结电容仍用下式计算

$$C_p = C_j A_{bot} + C_{jsw} P \tag{6.117}$$

但要记住,由于 nFET 和 pFET 的掺杂不同,它们的 C_j 和 C_{jsw}的数值也不同。pFET 的线性 RC 模型与图 6.27 中 nFET 的一样,只是要用 pFET 的值和一个低电平控制开关。

6.5 小尺寸 MOSFET 模型

本章介绍的公式是简化的模型,它对于一开始设计时的估算是非常有用的。它在 L 大于 20～30 μm 左右的**长沟道** MOSFET 中还是比较准确的;在分立(一个个分开的)器件中仍能找到它们。现代 IC 技术已将生产线 VLSI 晶体管的沟道长度缩小到 $L = 0.13$ μm,而且这一值仍在继续减小。这种亚微米尺寸器件的物理现象是相当复杂的,不可能找到贴切的表达式精确地描述这些晶体管。在电路设计层次,代之以两个层次的建模方法:尺寸缩小原理和计算机模拟。

6.5.1 尺寸缩小原理

尺寸缩小原理涉及"超乎想像地缩小晶体管",并指导我们去研究当一个器件的尺寸按结构方式缩小时它的行为。

考虑一个沟道宽度为 W,沟道长度为 L 的晶体管。我们希望发现,当这两个尺寸都按同一个缩小因子 $s>1$ 缩小时其主要电学特性是如何变化的。这时,新的(被缩小的)晶体管的尺寸为

$$\tilde{W} = \frac{W}{s} \qquad \tilde{L} = \frac{L}{s} \tag{6.118}$$

我们注意到,原有晶体管的栅面积为 $A = WL$,而缩小后的 FET 的面积为

$$\tilde{A} = \frac{A}{s^2} \tag{6.119}$$

例如，$s=2$ 意味着缩小的器件只占原有面积的25%。这就为继续改进光刻工艺提供了巨大的推动力。

让我们看一下器件的互导。既然 W 和 L 都被缩小同样的倍数，所以它们的宽长比不变：

$$\left(\frac{W}{L}\right)=\left(\frac{\tilde{W}}{\tilde{L}}\right) \tag{6.120}$$

氧化层电容为

$$C_{ox}=\frac{\varepsilon_{ox}}{t_{ox}} \tag{6.121}$$

式中 t_{ox} 是栅氧层厚度。如果新的FET的氧化层较薄，即缩小为

$$\tilde{t}_{ox}=\frac{t_{ox}}{s} \tag{6.122}$$

则缩小的器件有

$$\tilde{C}_{ox}=\frac{\varepsilon_{ox}}{\left(\frac{t_{ox}}{s}\right)}=sC_{ox} \tag{6.123}$$

即它增加了 s 倍。由于工艺互导为 $k'=\mu C_{ox}$，那么在缩小的器件中器件互导 $\beta=k'(W/L)$ 增加为

$$\tilde{\beta}=sk'\left(\frac{W}{L}\right)=s\beta \tag{6.124}$$

然而注意，L 和 W 能按 s 倍缩小并不意味着氧化层的厚度也可以缩小同样的倍数，所以在应用这一关系时一定要小心。如果确实可以，则FET电阻

$$R=\frac{1}{\beta(V_{DD}-V_T)} \tag{6.125}$$

可以缩小为

$$\tilde{R}=\frac{1}{s\beta(V_{DD}-V_T)} \tag{1.126}$$

如果不改变尺寸缩小后FET的外加电压，则根据

$$\tilde{R}=\frac{R}{s} \tag{6.127}$$

电阻就会降低。反之，如果能在这个小器件中将电压降低到一个新的值，即

$$\tilde{V}_{DD}=\frac{V_{DD}}{s},\qquad \tilde{V}_T=\frac{V_T}{s} \tag{6.128}$$

则缩小的FET的电阻将不变，即

$$\tilde{R}=R \tag{6.129}$$

这就是**电压缩小**的基础，即当器件尺寸缩小时，也降低电压。

要了解降低电压的影响，考虑一个缩小的MOSFET，它的电压降低为

$$\tilde{V}_{GS}=\frac{V_{DS}}{s},\qquad \tilde{V}_{GS}=\frac{V_{GS}}{s} \tag{6.130}$$

原有器件的不饱和电流为

$$I_D=\frac{\beta}{2}[2(V_{GS}-V_T)V_{DS}-V_{DS}^2] \tag{6.131}$$

利用缩小公式,得到在缩小的 FET 中的电流为

$$\tilde{I}_D=\frac{s\beta}{2}\left[2\left(\frac{V_{GS}}{s}-\frac{V_T}{s}\right)\frac{V_{DS}}{s}-\frac{V_{DS}^2}{s^2}\right]=\frac{I_D}{s} \tag{6.132}$$

晶体管的功耗为

$$\tilde{P}=\tilde{V}_{DS}\tilde{I}_D=\frac{V_{DS}I_D}{s^2} \tag{6.133}$$

即减少了 $1/s^2$ 倍。这就是为什么当 FET 尺寸缩小时也希望降低电源电压的原因。

电源电压的实际值 V_{DD}是在系统级决定的,经常用它来降低电路的功耗。阈值电压值 V_T 由工艺控制。虽然工作电压可以做某些改变,但工作电压的减少通常与几何尺寸缩小 s 倍不同。然而,这确实告诉了我们,什么是可以期望得到的。

6.5.2　小尺寸器件效应

随着 20 世纪 80 和 90 年代 MOSFET 尺寸的缩小,很自然地要对电流公式提出修正,以便解释最新观察到的效应。许多新一类的现象被发现且研究,而且以前的许多术语和词汇至今仍在沿用。

在 VLSI FET 中最重要的几何参数是沟道长度 L。由于宽长比(W/L)决定了通过晶体管的最大电流,缩小 L 时可以同时缩小 W 而仍保持同样的宽长比。在下一章中将说明宽长比是基本的电路设计参数。缩小的电路因此消耗较少的面积,但仍保持一些重要的电路特性不变。

当沟道长度减小到 20 μm 以下时,发现阈值电压会从它的长沟道值 $V_{T,long}$降下来。这称为**几何短沟效应**(SCE),并且可用下列公式表示为

$$V_T=V_{T,long}-(\Delta V_T)_{SCE} \tag{6.134}$$

式中$(\Delta V_T)_{SCE}$随 L 的减小而增加。阈值电压的减少量可以通过比在长沟道的推导中更精确地考虑电荷来计算。**窄沟道宽度效应**(NWE)也是一个考虑几何的修正量,即阈值电压随着 W 的缩小而增大。这可用下式表示为

$$V_T=V_{T,long}+(\Delta V_T)_{NWE} \tag{6.135}$$

它来源于在长沟道分析中忽略不计的边缘电场。最小尺寸器件可以同时表现出 SCE 和 NWE 效应。

L 的缩小也引起沟道导电特性的改变。考虑一个具有外加漏源电压 V_{DS}的 FET,其沟道电场可以估计为

$$E \approx \frac{V_{DS}}{L} \tag{6.136}$$

上式说明 E 随着 L 的缩小而增加。我们观察到,在硅中一个带电粒子的速度符合图 6.35 所示的关系。当 E 值较小时,速度按下式关系线性增加

$$v = \mu E \tag{6.137}$$

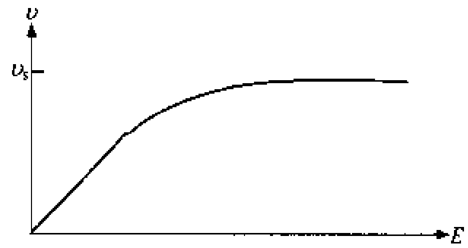


图 6.35 硅中带电粒子的速度与电场的关系

它定义了在推导 FET 电流公式时所用的迁移率 μ。然而,随着电场强度的增加,$v(E)$进入非线性区,因而迁移率也不再是一个常数。它的值最终达到**饱和速度** v_s,室温下硅中的电子饱和速度约为 10^7 cm/s。因此简化的公式不再适用,必须进行修正。现代短沟道 FET 通常都进入速度饱和区。

速度对于估算渡越时间 τ_t 也是很有用的,τ_t 是一个电荷横越沟道所需要的时间。这被认为是对 FET 开关速度快慢的基本限制,用速度 v 写出下列公式

$$\tau_t = \frac{v}{L} \tag{6.138}$$

在 v-E 曲线的线性区

$$\tau_t = \frac{\mu E}{L} = \frac{\mu V_{DS}}{L^2} \tag{6.139}$$

这是缩小沟道长度的又一理由:因为 τ_t 值降低了,这标志着开关速度加快。一旦粒子速度达到饱和,渡越时间成为常数

$$\tau_t = \frac{v_s}{L} \tag{6.140}$$

其影响就不再那么显著了。短沟道的另一效应是电流仅涉及少量的电子。许多与统计推导电荷浓度有关的假设开始不再成立。

在小几何尺寸 MOSFET 中还观察到许多其他效应,随着器件尺寸的继续缩小和新的晶体管的不断出现和开发,研究正在继续进行。有兴趣的读者可直接参阅近期文献,了解更详细的情况。

6.5.3 SPICE 模型

多年来我们已经明白,不可能推导出精确描述现代晶体管的贴切公式。幸运的是,成熟 CAD 工具的开发使我们能在器件和电路层次上进行精确的模拟。器件模拟器不在本书介绍的范围之内。另一方面,电路模拟是 VLSI 电路设计的一个常规过程。设计流程首先用 FET 的开关理论建立逻辑电路,再用简化的方程估算电气特性。然后进行电路模拟,并将其结果用来改进电气设计。使用最广泛的电路模拟工具是 SPICE。⑤ 这一程序由美国加州 Berkeley 大学提出设想并完成,以帮助集成电路的设计。由于 SPICE 已成为工业界的标准,我们将把讨

⑤ 这是英文 Simulation Program with Integrated Circuit Emphasis 的缩写。

论集中在它上面。SPICE 已有几个版本，但它们的运行都很类似。

在 SPICE 中 MOSFET 是用如下形式的元件语句列出在电路表中的：

```
Mname ND NG NS NB model _ name L=length W=width<AS, PS, AD, PD>
```

其中

- Mname 是 FET 的名字，例如 M1 或 Mn _ out
- ND, NG, NS, NB 分别为漏，栅，源和体的节点号码(如果允许的话也可以用名字表示)
- model _ name 是 .model 这一行的名字，它列出了工艺参数
- AS, PS, AD, PD 是器件漏(AD, PD)和源(AS, PS)的(可选)面积和周长。面积单位为 m^2，周长必须用米(m)单位来说明

重要的电气参数包括在 .model 行中，其形式为

```
.model＜参数表＞
```

这里＜参数表＞是数值表。现在已有许多 MOSFET 模型。在＜参数表＞中用语句

```
Level = N
```

来区分它们，N 的值确定所用的方程组。最初的 SPICE 有 1,2,3 三个 Level。其中 Level 1 是基于 6.2.3 小节推导的方程组进行修改后的形式；Level 2 模型也叫体电荷方程，是更为精确的模型；而 Level 3 是一个经验模型。Level 1 和 2 应用在现代亚微米器件中时不够精确，但常用于最初的估计，因为它们可以进行非常快速的模拟。

在元件语句中的所有尺寸通常都输入设计值(所画的值)。例如，例 6.6 中的晶体管可以描述为

```
Mexa6 _ 6 10 20 30 0 nFET L=0.4U W=5U AD=15P PD=16U AS=15P PS=16U
```

其中，P 表示微微，U 表示微。设计值与有效(出于电气考虑)值之间的差是通过如下工艺信息来进行计算的：

```
.MODEL nFET ＜参数＞
```

这使由版图转为模拟文件容易得多。

在现代 CMOS 中，BSIM 模型提供最精确的 SPICE 模拟。[7] 可惜的是，参数本身有一定保密程度，并且它们的值并不总是与简单的解析表达式有直接的联系。关于 BSIM 模型的细节可参阅参考文献[2]。在 VLSI 设计中，通常把模型理解为一组能在 CAD 工具组中使用的给定参数。根据版图中提取的网表可以用来进行电路模拟。

6.6　参考资料

[1] R. Jacob Baker, Harry Li, and David E. Boyce, **CMOS Circuit Design, Layout and, Simulation,** IEEE Press, Piscataway, NJ, 1998.

---

⑦ BSIM 代表 Berkeley Submicron IGFET Model，IGFET 代表绝缘栅 FET(Insulated Gate FET)。在日常使用中，IGFET 和 MOSFET 可互相替代使用。

[2] Yuhua Cheng and Chenming Hu, **MOSFET Modeling and BSIM3 User's Guide**, Kluwer Academic Press, Norwell, MA, 1999.

[3] Richard S. Muller and Theodore I. Kamins, **Device Electronics for Integrated Circuits**, 2nd ed., John Wiley & Sons, New York, 1992.

[4] Robert F. Pierret, **Semiconductor Device Fundamentals**, Addison-Wesley, Reading, MA, 1996.

[5] Ben G. Streetman and Sanjay Banerjee, **Solid State Electronic Devices**, 5th ed., Prentice Hall, Upper Saddle River, NJ, 1999.

[6] Jasprit Singh, **Semiconductor Devices**, John Wiley & Sons, New York, 2001.

[7] S. M. Sze, **Semiconductor Devices**, 2nd ed., Wiley-Interscience, New York, 1981.

[8] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[9] Edward S. Yang, **Microelectronic Devices**, McGraw-Hill, New York, 1988.

6.7 习题

[**6.1**] 有一 CMOS 工艺生产的栅氧层厚度为 $t_{ox}=10$ nm。已知 FET 载流子的迁移率为 $\mu_n=550\ \text{cm}^2/(\text{V}\cdot\text{s})$，$\mu_p=210\ \text{cm}^2/(\text{V}\cdot\text{s})$。

(a) 计算单位面积的氧化层电容，单位用 $\text{fF}/\mu\text{m}^2$

(b) 求 nFET 和 pFET 的工艺互导值，单位用 $\mu\text{A/V}^2$

[**6.2**] 一个 nFET 的 $W=10\ \mu m$，$L=0.35\ \mu m$，其所用工艺为 $k'_n=110\ \mu\text{A/V}^2$，$V_{Tn}=0.70$ V。假设 $V_{SBn}=0$ V。

(a) 求电压设定为 $V_{GSn}=2$ V，$V_{DSn}=1.0$ V 时的电流。

(b) 求电压设定为 $V_{GSn}=2$ V，$V_{DSn}=2$ V 时的电流。

[**6.3**] 一个 nFET，它的器件互导为 $\beta_n=2.3\ \text{mA/V}^2$，阈值电压为 0.76 V。假设 $V_{SBn}=0$ V。

(a) 求电压设定为 $V_{GSn}=1$ V，$V_{DSn}=2.5$ V 时的电流。

(b) 求电压设定为 $V_{GSn}=2$ V，$V_{DSn}=2.5$ V 时的电流。

(c) 求电压设定为 $V_{GSn}=3$ V，$V_{DSn}=2.5$ V 时的电流。

[**6.4**] 有一 pFET，其栅氧化层厚度为 $t_{ox}=6$ nm。测得其空穴迁移率为 220 $\text{cm}^2/(\text{V}\cdot\text{s})$，宽长比为 $(W/L)=(12/1)$。假设 $V_{DD}=3.3$ V，$|V_{Tp}|=0.7$ V。

(a) 计算工艺互导 k'_p，单位用 mA/V^2。

(b) 求器件互导 β_p 和电阻 R_p。

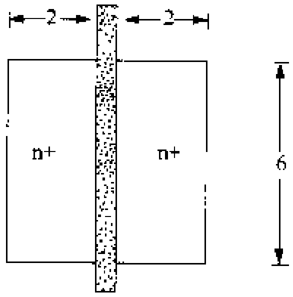
[**6.5**] 一个 nFET，栅氧化层厚度为 $t_{ox}=12$ nm。其 p 型体区域掺硼密度为 $N_a=8\times10^{14}\text{cm}^{-3}$。已知 $V_{T0n}=0.55$ V，$(W/L)=10$。

(a) 计算体偏置系数 γ。

(b) 如果外加体偏置电压 $V_{SBn}=2$ V，器件的阈值电压应为多少？

(c) 已知电子迁移率为 $\mu_n = 540\ cm^2/(V\cdot s)$，求器件偏置电压为 $V_{GSn}=3\ V$，$V_{DSn}=3\ V$，$V_{SBn}=3\ V$ 时的漏电流

[6.6] 为图 P6.1 中的 FET 版图建立 RC 开关模型。假设电源电压为 3 V，尺寸单位为微米。



$L'=0.5\ \mu m$　　$k_n'=150\ \mu A/V^2$
$L_o=0.05\ \mu m$　　$C_{ox}=2.70\ fF/\mu m^2$
$V_{Tn}=0.6\ V$　　$C_j=0.86\ fF/\mu m^2$
$C_{jsw}=0.24\ fF/\mu m$

图 P6.1 题 6.6 中晶体管版图的几何图

[6.7] 写出图 P6.1 中 nFET 的 SPICE 描述。用你列出的参数表得到 I_D 与 V_{DS} 的关系的曲线族。

[6.8] 考虑图 P6.1 中的 FET 几何图，n+ 区的薄层电阻为 30 Ω，多晶栅的薄层电阻为 26 Ω。通过确定寄生电阻 R_{n+} 和 R_{poly} 各自合适的几何尺寸以及相关的薄层电阻参数计算它们的电阻值。这些寄生电阻是如何影响器件工作的？

[6.9] 一个 nFET 的 $W=20\ \mu m$，$L=0.5\ \mu m$，采用 $k_n'=120\ \mu A/V^2$ 及 $V_{Tn}=0.65\ V$ 的工艺制造。若设电压为 $V_{GSn}=V_{DSn}=V_{DD}=5\ V$。

(a) 该管子处于饱和还是非饱和状态？

(b) 采用该管子合适的方程计算它的漏源电阻。

(c) 将你在(b)中计算出的值与 $\eta=1$ 时利用方程(6.71)计算出来的值相比较。

[6.10] 一个 nFET 的 $L=0.5\ \mu m$，采用 $k_n'=100\ \mu A/V^2$ 及 $V_{Tn}=0.70\ V$ 的工艺制造。栅源电压设为 $V_{GSn}=V_{DD}=3.3\ V$。为使电阻 $R_n=950\ \Omega$，试用方程(6.71)计算当 $\eta=1$ 时所应有的沟道宽度。

第 7 章 CMOS 逻辑门电子学分析

前一章我们考察了 MOSFET 的电气特性。这为本章分析在 CMOS 逻辑电路中晶体管的行为建立了基础。本章将集中阐述开关速度和版图设计的重要问题,且为现代芯片设计的许多方面提供基础。

7.1 CMOS 反相器的直流特性

CMOS 反相器是计算逻辑门电气特性的基础。考虑图 7.1 中的电路,其输入电压 V_{in}决定了两个 FET Mn 和 Mp 的导电状态。由此产生了门的输出电压 V_{out}。描述一个数字逻辑电路特性需要两种类型的计算。**直流(DC)分析**确定了对于一个给定输入值 V_{in}时的输出值 V_{out}。在这类计算中,假设 V_{in}的变化非常慢,并且 V_{out}可以在测量前处于稳定状态。DC 分析提供了输入至输出间的直接映射,从而告诉我们定义布尔逻辑值 0 和 1 的电压范围。第 2 种类型的特性分析叫做**暂态分析**。这时,输入电压相应于变化的逻辑值为时间的显函数 $V_{in}(t)$。电路的响应包括在 $V_{out}(t)$中。在输入变化和相应的输出变化之间的延时是高速设计的基本限制因素。本节将集中讨论 DC 分析。暂态反应分析将在下一节进行。

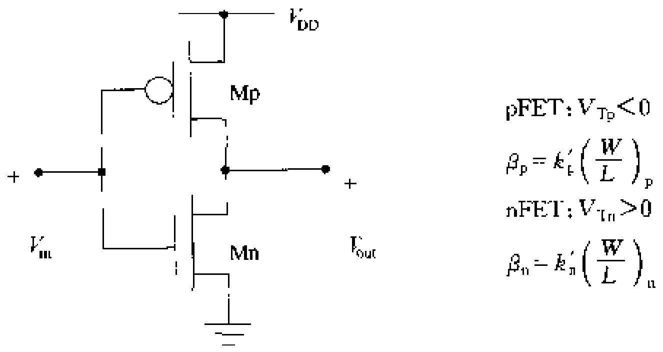


图 7.1 CMOS 反相器电路

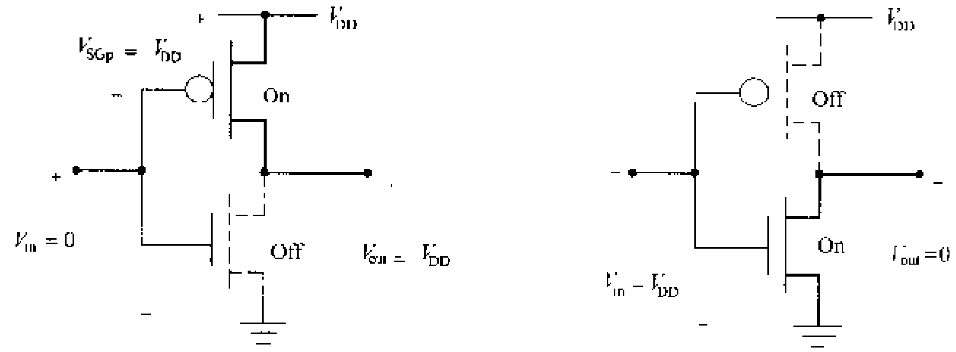
反相器的 DC 特性可由**电压传输特性(VTC)**来说明,这是 V_{out}作为 V_{in}函数的曲线。这条曲线通过在 0 V 到 V_{DD}范围内改变输入电压 V_{in}并且求出输出电压 V_{out}得到。靠着图 7.2 中电路的帮助,可以很容易地找到两端的值。如图 7.2(a)所示,如果 V_{in}等于 0 V,则 Mn 截止而 Mp 导通。由于 pFET 导通,使输出与电源相连,于是 $V_{out}=V_{DD}$。这就确定了该电路**输出高电压**

$$V_{OH}=V_{DD} \tag{7.1}$$

即最高输出电压是电源电压 V_{DD}。反之如图 7.2(b),当 $V_{in}=V_{DD}$时,它使 Mn 导通而 Mp 截

止，于是输出节点通过 nFET 与 0 V(地)相连，这就确定了**输出低电压**

$$V_{OL} = 0\ \text{V} \tag{7.2}$$



(a) 低输入电压　(b) 高输入电压

图 7.2　反相器电路的 V_{OH}和 V_{OL}

输出端的**逻辑摆幅**为

$$\begin{aligned} V_L &= V_{OH} - V_{OL} \\ &= V_{DD} \end{aligned} \tag{7.3}$$

由于它等于全部电源电压值，所以叫做**全轨输出**(full-rail output)。

电路的 VTC 是使输入电压值从 $V_{in}=0$ V 开始然后逐渐增加到 $V_{in}=V_{DD}$获得的，所得到的曲线图见图 7.3。详细情况可以通过写出器件的输入和输出电压来理解：

$$\begin{aligned} V_{GSn} &= V_{in} \\ V_{SGp} &= V_{DD} - V_{in} \end{aligned} \tag{7.4}$$

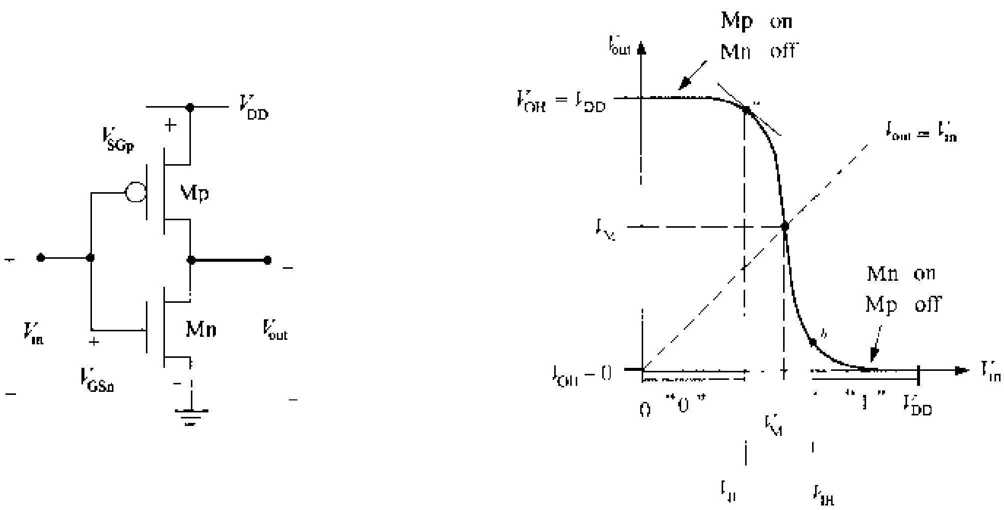


图 7.3　NOT 门的电压传输曲线

只要 $V_{in} \leqslant V_{Tn}$，Mn 就处于截止状态。由于此时输出电压为高电压值 $V_{out}=V_{DD}$，因此在标为"0"范围内的任何输入电压都可看成一个逻辑 0 的输入。增加 V_{in}使 VTC 下行，这是因为输入电压使 nFET 导通而此时 pFET 还仍然在导电。但是注意，增加 V_{in}使 V_{SGp}减小，所以

pFET 的导电性变弱，使输出电压下降。当

$$V_{in} = V_{DD} - |V_{Tp}| \tag{7.5}$$

时，Mp 截止。如果 V_{in}大于该值，由于只有 nFET 起作用，所以 $V_{out} = 0$ V。这说明输入电压在某个范围内的作用相当于一个逻辑 1 的输入值，如图 VTC 上"1"所标明的部分。

逻辑 0 和逻辑 1 的电压范围由 VTC 的斜率决定。图中的斜率是变化的，在点 a 处的斜率为 −1，它确定了**输入低电压** V_{IL}。根据定义，一个逻辑 0 的输入电压为

$$0 \leqslant V_{in} \leqslant V_{IL} \tag{7.6}$$

第 2 个 −1 斜率点标为 b，它确定了**输入高电压** V_{IH}。由它可以定义逻辑 1 的输入电压为

$$V_{IH} \leqslant V_{in} \leqslant V_{DD} \tag{7.7}$$

高电压和低电压的**电压噪声容限**分别为

$$\begin{aligned} VNM_H &= V_{OH} - V_{IH} \\ VNM_L &= V_{IL} - V_{OL} \end{aligned} \tag{7.8}$$

噪声容限是输入对于电磁耦合信号干扰稳定性的定量度量。

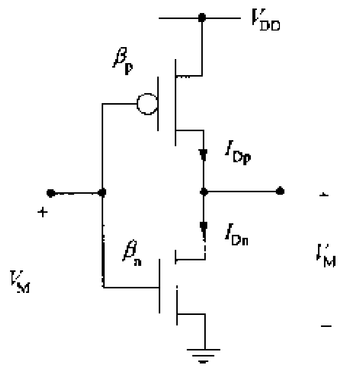


图 7.4 用来计算 V_M 的反相器电压

虽然能够算出定义逻辑 0 和逻辑 1 输入电压的确切值，若在 VTC 中所示引入一个中点电压 V_M 则会更简单些。该点定义为 VTC 与 $V_{out} = V_{in} = V_M$ 所确定的单位增益线的交点。$V_{in} = V_M$ 的值位于过渡区因而并不代表一个布尔量。但是，当 V_{in}小于 V_M 时输入电压趋于逻辑 0，而 $V_{in} > V_M$ 时则标志着输入处于逻辑 1 这边。因此知道了 V_M 的值就知道了输入过渡变化的中点。

为了计算中点电压，在图 7.4 中先设 $V_{in} = V_{out} = V_M$。使两个 FET 的漏电流相等得到：

$$I_{Dn} = I_{Dp} \tag{7.9}$$

但在能够使用该表达式之前必须确定每个 FET 的工作区域（饱和或非饱和的）。首先考虑 nFET 并且回想一下前面的饱和电压公式，即

$$\begin{aligned} V_{sat} &= V_{GSn} - V_{Tn} \\ &= V_M - V_{Tn} \end{aligned} \tag{7.10}$$

在第 2 行中我们利用了 $V_{in} = V_{GSn} = V_M$。漏源电压为 $V_{DSn} = V_{out} = V_M$。由于 V_{Tn}是一个正数，所以

$$V_{DSn} > V_{sat} = V_M - V_{Tn} \tag{7.11}$$

这说明 Mn 必定饱和。由于 $V_{SGp} = V_{SDp}$，同样的理由也适用于 pFET Mp。由第 6 章中的饱和电流公式得到

$$\frac{\beta_n}{2}(V_M - V_{Tn})^2 = \frac{\beta_p}{2}(V_{DD} - V_M - |V_{Tp}|)^2 \tag{7.12}$$

两边除以 β_p 并取平方根得到

$$\sqrt{\frac{\beta_n}{\beta_p}}(V_M - V_{Tn}) = V_{DD} - V_M - |V_{Tp}| \tag{7.13}$$

经简单的代数运算得到中点电压为

$$V_M = \frac{V_{DD} - |V_{Tp}| + \sqrt{\frac{\beta_n}{\beta_p}} V_{Tn}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{7.14}$$

该公式表明 V_M 是由 nFET-pFET 之比确定：

$$\frac{\beta_n}{\beta_p} = \frac{k'_n\left(\frac{W}{L}\right)_n}{k'_p\left(\frac{W}{L}\right)_p} \tag{7.15}$$

由于 k'_n 和 k'_p 均在工艺过程中设定，所以 FET 的尺寸比确定了切换点。切记，nFET 和 pFET 具有不同的迁移率，依工艺细节不同它们的典型比值为

$$\frac{k'_n}{k'_p} \approx 2 \sim 3 \tag{7.16}$$

这个事实对于在复杂的 VLSI 设计中选择各个晶体管的尺寸以及使用的电路类型，都具有重要意义。注意，由于两种 FET 类型的 C_{ox} 近似相等，

$$\frac{k'_n}{k'_p} = \frac{\mu_n}{\mu_p} = r \tag{7.17}$$

式中 r 是第 5 章介绍过的迁移率之比。

一个**对称反相器**的 VTC 是一个“0”和“1”的输入电压范围相同的 VTC。可以令公式(7.12)中的

$$V_M = \frac{1}{2}V_{DD} \tag{7.18}$$

来得到这种 VTC。经整理后得到设计方程

$$\frac{\beta_n}{\beta_p} = \left(\frac{\frac{1}{2}V_{DD} - |V_{Tp}|}{\frac{1}{2}V_{DD} - V_{Tn}}\right)^2 \tag{7.19}$$

它可计算 V_M 为这一特别值时晶体管的尺寸。注意，如果 $V_{Tn} = |V_{Tp}|$，则对称设计要求

$$\beta_n = \beta_p \tag{7.20}$$

即两个 FET 器件的互导值相等。应当记住，β 与 MOSFET 的宽长比(W/L)成正比，且(W/L)是实际的设计变量。

【例 7.1】 一个 CMOS 工艺具有下列参数：

$$k'_n = 140\ \mu A/V^2 \qquad V_{Tn} = +0.70\ V$$
$$k'_p = 60\ \mu A/V^2 \qquad V_{Tp} = -0.70\ V \tag{7.21}$$
$$V_{DD} = 3.0\ V$$

考虑 $\beta_n = \beta_p$ 的情况。通过下列计算可以验证这是一个对称设计：

$$V_M = \frac{3 - 0.7 + \sqrt{1}(0.7)}{1 + \sqrt{1}} = 1.5\ V \tag{7.22}$$

所以 V_M 是电源电压值的一半。要实现这一设计，我们必须选择器件的宽长比为：

$$\frac{\beta_n}{\beta_p} = \frac{k'_n\left(\frac{W}{L}\right)_n}{k'_p\left(\frac{W}{L}\right)_p} = 1 \tag{7.23}$$

这里，回想一下工艺互导参数为 $k' = \mu_n C_{ox}$，并且这是由工艺过程设定的。就这一情形，将(7.23)式整理后得到

$$\left(\frac{W}{L}\right)_p = \frac{k'_n}{k'_p}\left(\frac{W}{L}\right)_n \tag{7.24}$$

所以

$$\left(\frac{W}{L}\right)_p = \left(\frac{140}{60}\right)\left(\frac{W}{L}\right)_n = 2.33\left(\frac{W}{L}\right)_n \tag{7.25}$$

它表明 pFET 的大小必须为 nFET 的 2.33 倍左右。

让我们来考察一下 nFET 和 pFET 具有同样宽长比 $(W/L)_n = (W/L)_p$ 的情况。由上面讨论得到的值

$$\frac{\beta_n}{\beta_p} = \frac{k'_n}{k'_p} = 2.33 \tag{7.26}$$

所以中点电压为

$$V_M = \frac{3 - 0.7 + \sqrt{2.33}\ (0.7)}{1 + \sqrt{2.33}} = 1.33\ V \tag{7.27}$$

这一选择使 V_M 的值变为小于 $(V_{DD}/2)$。

图 7.5 表明一个反相器采用两种设计类型时版图上的区别。反相器中两个晶体管的沟道长度相等，沟道宽 W_p 和 W_n 则为设计变量。在图 7.5(a)中，pFET 的沟道宽度约为 $W_p \approx 2W_n$，所以 V_M 约为 $V_{DD}/2$。图 7.5(b)中的版图采用了相同尺寸的晶体管，所以电路的 $V_M < (V_{DD}/2)$。记住目前我们只关心 DC 的特性，下一节将看到这两个设计的开关特性同样要受尺寸比的影响。

上述推导及例子表明在逻辑门的 DC 行为中 FET 宽长比的重要性。在物理层次上，器件的相对尺寸体现在比例(β_n/β_p)中，它决定了切换点。总的来说，增大(β_n/β_p)将降低中点电压 V_M 的值。这一相关性表现在图 7.6 的曲线图上。如所示的参数，由 $\beta_n=\beta_p$ 的对称设计得到 $V_M=(V_{DD}/2)=1.5$ V。将比值加大到(β_n/β_p)= 1.5，则 $V_M\approx1.42$ V；而(β_n/β_p)= 2.5 时，中点电压降低为 $V_M\approx1.31$ V。也可以使(β_n/β_p)< 1，则 VTC 右移，即 $V_M>(V_{DD}/2)$。不过很少这样做，因为这样会使 pFET 的宽长比变得太大。



(a) 尺寸较大的 pFET 设计　　(b) 相同宽长比

图 7.5　例 7.1 中两种设计的版图比较

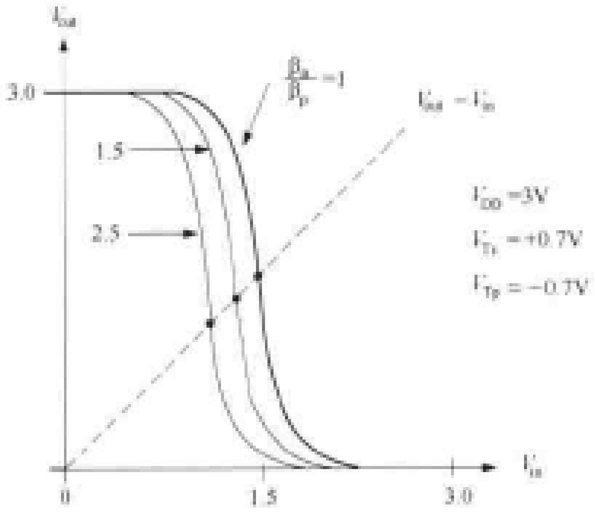


图 7.6　V_M 与器件比的关系

7.2　反相器的开关特性

高速数字系统的设计是以进行快速计算的能力为基础的。这就要求在输入改变时逻辑门引起的时延最小。设计快速逻辑电路是 VLSI 物理设计比较有挑战性(但也是很关键的)的一个方面。与 DC 分析一样分析非门可为研究更复杂的电路提供基础。

图 7.7 显示了该问题的一般特点。在反相器上加上输入电压 $V_{in}(t)$，得到输出电压 $V_{out}(t)$。我们假设 $V_{in}(t)$ 具有阶梯形特点，即在时间 t_1 时从 0 突变到 1(即电压 V_{DD})，然后在

时间 t_2 时又降回到 0。输出波形会对输入做出响应，但输出电压不可能瞬时改变。输出 1 至 0 的过渡会有一个**下降时间**延时 t_f，而输出 0 到 1 的变化则可用**上升时间** t_r 来描述。上升和下降时间可以通过电路过渡的电子学分析来计算。

上升和下降时间延迟是由于晶体管的寄生电阻和电容造成的。考虑图 7.8(a)中的非门电路。两个 FET 都可用它们的等效开关来代替，得到图 7.8(b)中简化的 RC 模型。值得回想一下等效电路中元件的实际值取决于器件的尺寸。一旦确定了宽长比，就可以用下式计算 R_n 和 R_p：

$$R_n = \frac{1}{\beta_n(V_{DD} - V_{Tn})}$$
$$R_p = \frac{1}{\beta_p(V_{DD} - |V_{Tp}|)} \tag{7.28}$$

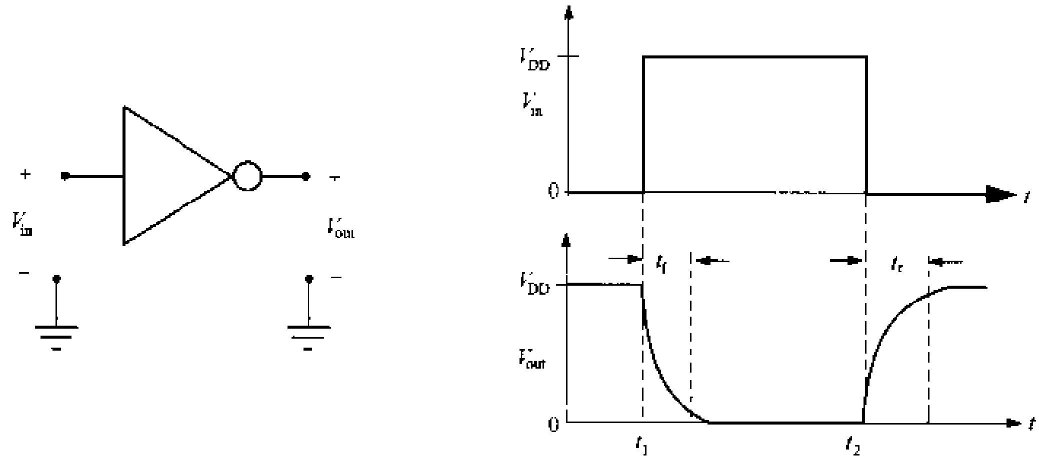


图 7.7　一般的开关波形

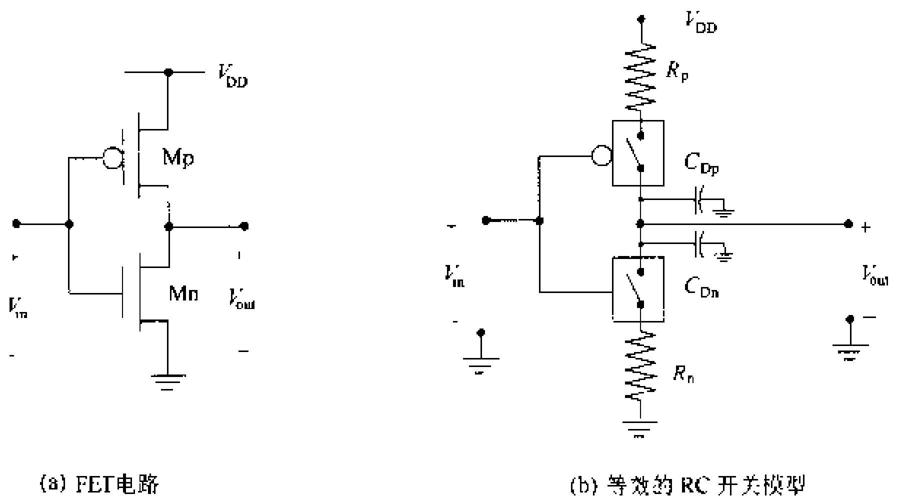


图 7.8　CMOS 反相器的等效 RC 开关模型

知道了每个 FET 的版图尺寸，就可求出输出节点上的电容 C_{Dn}和 C_{Dp}。计算公式为：

$$C_{Dn} = C_{GSn} + C_{DBn} = \frac{1}{2}C_{ox}L'W_n + C_{jn}A_n + C_{jswn}P_n$$
$$C_{Dp} = C_{GSp} + C_{DBp} = \frac{1}{2}C_{ox}L'W_p + C_{jp}A_p + C_{jswp}P_p \quad (7.29)$$

式中分别用下标 n 和 p 来区分 nFET 或 pFET 的各个量。[①] 记住，增加一个 FET 的沟道宽度将会加大寄生电容的值，这一点非常重要。

在得到一个完整的模型之前，还有一点十分重要，这就是在逻辑链中每个逻辑门必须能够驱动另一个门或一组门，这才有用。所驱动的门的数目是由电路的**扇出数**(FO)确定的。所谓扇出的门就是驱动电路的**负载**，因为它们都存在着**输入电容** C_{in}。考虑图 7.9 (a)中的反相器，反相器的输入电容就是两个 FET 的电容之和

$$C_{in} = C_{Gp} + C_{Gn} \quad (7.30)$$

图 7.8(b)显示了一个扇出 FO=3 的输入电容的影响。输入电容对于每个门的作用如同一个驱动门的**外加负载电容** C_L。在本例中很容易看到

$$C_L = 3C_{in} \quad (7.31)$$

为加在非门上的负载值。

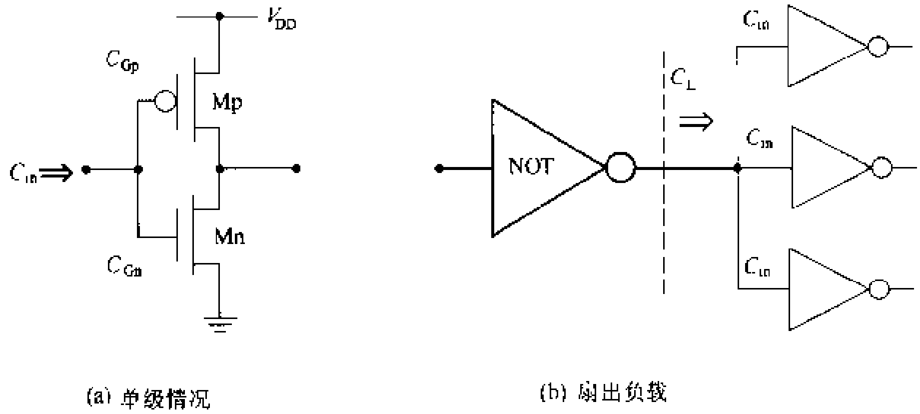


图 7.9 输入电容与负载效应

现在可以计算反相器的开关时间了。图 7.10 说明了这个一般性问题。如图 7.10 (a)所示，一个 CMOS 非门用来驱动一个外加负载电容 C_L。这就得到了图 7.10 (b)中的完整的开关模型，其总输出电容为

$$C_{out} = C_{FET} + C_L \quad (7.32)$$

前面图 7.8 中的 FET 电容已合并成一项：

$$C_{FET} = C_{Dn} + C_{Dp} \quad (7.33)$$

这是其无法消除的内部寄生电容。由于所有这些电容是并联的，所以将它们与 C_L 相加。总输出电容 C_{out}是驱动门必须驱动的负载；其数值随负载而改变。

① 注意源电容 C_{Sp}和 C_{Sn}并未在本问题中考虑，因为它们分别位于电源和地，因而具有不变的电压。

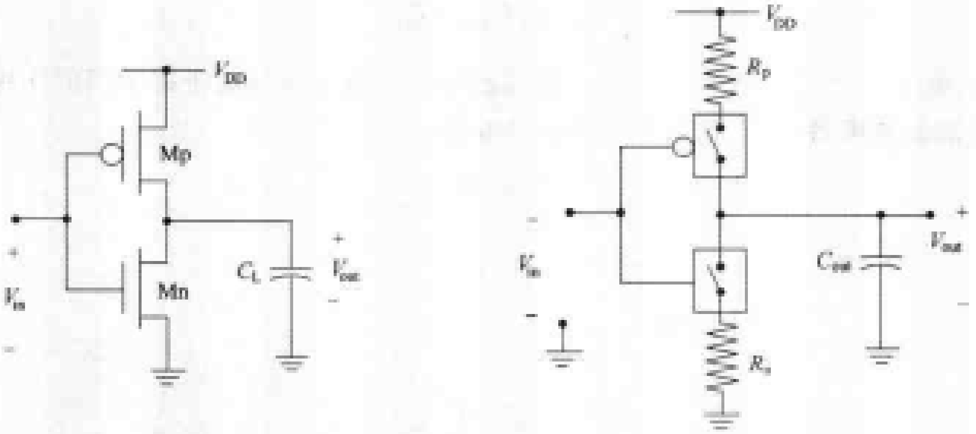
【例 7.2】 应用上述分析求图 7.11 中非门的电容。假设其中所有尺寸的单位均用微米(μm)表示。

首先求栅电容

$$
\begin{aligned}
C_{Gp} &= (2.70)(1)(8) = 21.6\ \text{fF} \\
C_{Gn} &= (2.70)(1)(4) = 10.8\ \text{fF}
\end{aligned}
\tag{7.34}
$$

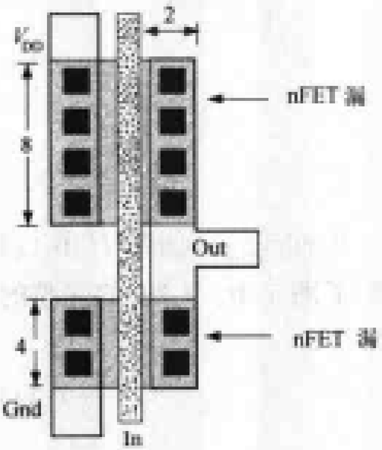
接着,注意重叠距离 L_o 为 0.1 μm,应当计入 pn 结电容的面积和周长的因子中。对于 pFET, p+电容为

$$C_p = C_j A_{bot} + C_{jsw} P_{sw} \tag{7.35}$$



(a) 外部负载　　(b) 完整的开关模型

图 7.10 完整的反相器开关模型的形成



$L = 1\ \mu$m(画出尺寸)
$L_o = 0.1\ \mu$m
$V_{DD} = 5$ V
$C_{ox} = 2.70$ fF/μm^2

pFET: $k_p' = 60\ \mu A/V^2$
$V_{T0p} = -0.7$ V
$C_j = 1.05$ fF/μm^2
$C_{jsw} = 0.32$ fF/μm^2

nFET: $k_n' = 150\ \mu A/V^2$
$V_{T0n} = 0.6$ V
$C_j = 0.86$ fF/μm^2
$C_{jsw} = 0.24$ fF/μm

图 7.11 电容计算举例

即

$$C_p = (1.05)(8)(2.1) + (0.32)2(8 + 2.1) = 24.10\ \text{fF} \tag{7.36}$$

所以 pFET 漏端的总电容应为

$$C_{Dp} = \frac{21.6}{2} + 24.10 = 34.9 \text{ fF} \tag{7.37}$$

用同样的步骤可以分析 nFET 的漏端。n+结电容为

$$C_n = (0.86)(4)(2.1) + (0.24)(2)(4+2.1) = 10.15 \text{ fF} \tag{7.38}$$

所以

$$C_{Dn} = \frac{10.8}{2} + 10.15 = 15.55 \text{ fF} \tag{7.39}$$

为 nFET 漏端的总电容。相加得到

$$\begin{aligned} C_{FET} &= C_{Dp} + C_{Dn} \\ &= 34.9 + 15.55 \\ &= 50.45 \text{ fF} \end{aligned} \tag{7.40}$$

为 FET 内部的总电容。输出端的总电容为

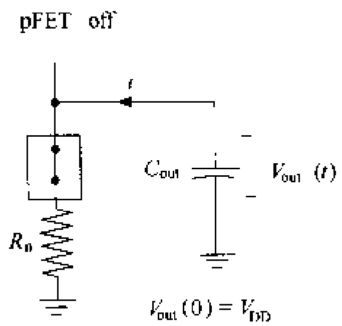
$$C_{out} = 50.45 + C_L \tag{7.41}$$

单位为 fF。其中 C_L 为外加负载(单位也是 fF)。

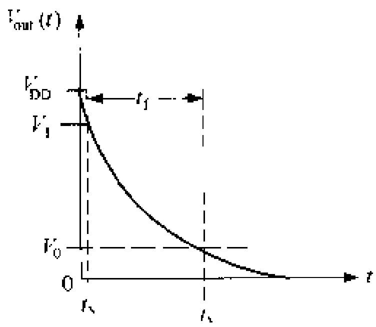
7.2.1　下降时间计算

让我们从计算输出下降时间 t_f 开始。将改变时间的起始点，即 V_{in}在时间 $t=0$ 时从 0 变为 V_{DD}。在输出端的初始情况是 $V_{out}(0)=V_{DD}$。当输入切换时，nFET 变为导通状态，而 pFET 则被驱动至截止。用开关模型来说，就是 nFET 开关闭合，pFET 开关断开。这样，得到的图 7.12(a)所示的简化放电电路。电容 C_{out}最初被充电到电压 V_{DD}。然后通过 nFET 电阻 R_n 放电至 0 V。离开电容的电流为

$$i = -C_{out}\frac{dV_{out}}{dt} = \frac{V_{out}}{R_n} \tag{7.42}$$



(a) 放电电路



(b) 输出波形

图 7.12　用于计算下降时间的放电电路

这是放电情况的微分方程。利用初始条件 $V_{out}(0)=V_{DD}$求解方程得到以下熟知的形式：

$$V_{out}(t) = V_{DD}e^{-t/\tau_n} \tag{7.43}$$

式中

$$\tau_n = R_n C_{out} \tag{7.44}$$

是单位为秒的**时间常数**。这个函数关系的曲线见图 7.12(b)。

一般下降时间定义为从 $V_1 = 0.9V_{DD}$到 $V_0 = 0.1V_{DD}$的时间间隔，这两个值分别称为全摆幅 V_{DD}的 90%和 10%电压。整理以上结果得到

$$t = \tau_n \ln\left(\frac{V_{DD}}{V_{out}}\right) \tag{7.45}$$

利用该式可以计算电压下降至某个特定电压 V_{out}所需要的时间 t。从图中可以看出

$$\begin{aligned} t_f &= t_y - t_x \\ &= \tau_n \ln\left(\frac{V_{DD}}{0.1V_{DD}}\right) - \tau_n \ln\left(\frac{V_{DD}}{0.9V_{DD}}\right) \\ &= \tau_n \ln(9) \end{aligned} \tag{7.46}$$

式中最后一步用到了恒等式

$$\ln(a) - \ln(b) = \ln\left(\frac{a}{b}\right) \tag{7.47}$$

取近似值 ln (9)≈2.2 得到电路下降时间的最后结果

$$t_f \approx 2.2\tau_n \tag{7.48}$$

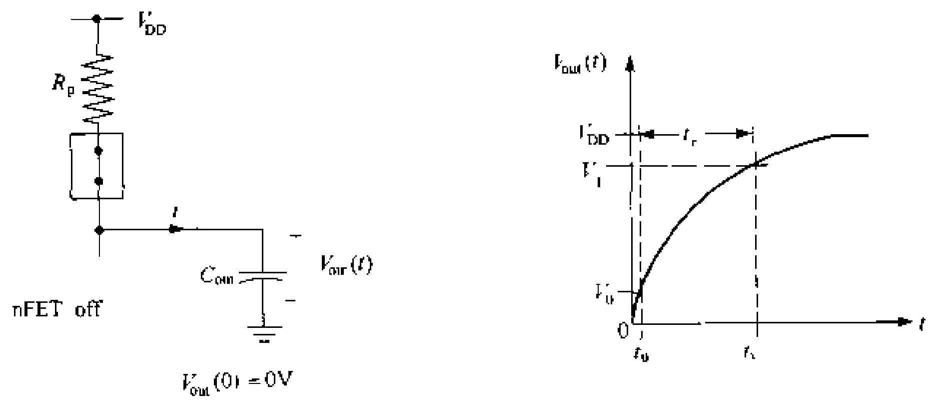
一般数字逻辑门中的输出下降时间通常称为输出端**高至低时间** t_{HL}，并且等于这里计算的值：

$$t_{HL} = t_f \tag{7.49}$$

在讨论中这两个符号将会交替使用。

7.2.2 上升时间

上升时间的计算可按同样方式。开始时输入电压为 $V_{in} = V_{DD}$，且被切换至 $V_{in} = 0$ V；为了简单起见，将发生这一情况的时间定为 $t = 0$。这使 pFET 导通而同时驱动 nFET 至截止，所以就有图 7.13 (a)中简化的充电电路。输出电压在 $t = 0$ 时为 $V_{out}(0) = 0$ V。



(a) 充电电路　　(b) 输出波型

图 7.13　上升时间计算

充电电流为

$$i = C_{out}\frac{dV_{out}}{dt} = \frac{V_{DD} - V_{out}}{R_p} \tag{7.50}$$

解方程并利用初始条件得到指数解：

$$V_{out}(t) = V_{DD}[1 - e^{-t/\tau_p}] \tag{7.51}$$

式中 pFET 的时间常数定义为

$$\tau_p = R_p C_{out} \tag{7.52}$$

图 7.13(b)显示输出电压为时间的函数。上升时间取为 10% 和 90% 点之间的时间，即

$$t_r = t_v - t_u \tag{7.53}$$

经代数运算得到上升时间 t_r 的表达式为：

$$t_r = \ln(9)\tau_p \approx 2.2\tau_p \tag{7.54}$$

这与下降时间 t_f 的形式相同，因为充电电路和放电电路是对称的。上升时间就是输出端的**低至高时间** t_{LH}；这两个符号也将被替换使用。

低至高时间 t_{LH} 和高至低时间 t_{HL} 分别代表输出电压从逻辑 0 到逻辑 1 或从逻辑 1 到逻辑 0 变化所需要的最短时间。假设输入为周期 T 秒的方波，电压在(T/2)的时间内为 0，而在另一个(T/2)的时间内为 V_{DD}。[②] 然后定义**最大信号频率**为

$$f_{max} = \frac{1}{t_{HL} + t_{LH}} = \frac{1}{t_r + t_f} \tag{7.55}$$

因为这是能够加在门上使输出仍可稳定至可确定状态的最大频率。[③] 如果信号频率超过 f_{max}，那么门的输出电压将没有足够的时间稳定到正确的值。

【例 7.3】 有一个反相器电路，FET 的宽长比 $(W/L)_n = 6$，$(W/L)_p = 8$。其工艺参数为

$$\begin{aligned} k'_n &= 150\ \mu A/V^2 \qquad V_{Tn} = +0.70\ V \\ k'_p &= 62\ \mu A/V^2 \qquad V_{Tp} = -0.85\ V \end{aligned} \tag{7.56}$$

电源电压 $V_{DD} = 3.3\,V$。总输出电容估计为 $C_{out} = 150\,fF$。让我们用前面推导的公式计算上升和下降时间。

首先考虑上升时间。pFET 的电阻为

$$\begin{aligned} R_p &= \frac{1}{\beta_p(V_{DD} - |V_{Tp}|)} \\ &= \frac{1}{(62\times10^{-6})(8)(3.3-0.85)} \\ &= 822.9\ \Omega \end{aligned} \tag{7.57}$$

② 这就是所谓的 50% 工作周期。

③ 该定义假设 t_{HL} 和 t_{LH} 具有相同的数量级时才有意义。

用 RC 乘积即 R_pC_{out}得到充电的时间常数为：

$$\tau_p = (822.9)(150\times10^{-15}) = 123.43 \quad \text{ps} \tag{7.58}$$

式中 1 ps(1 皮秒)为 10^{-12}秒。上升时间为

$$t_r = 2.2\tau_p = 271.55 \quad \text{ps} \tag{7.59}$$

下降时间可用类似的方法来计算。首先求 nFET 电阻

$$\begin{aligned} R_n &= \frac{1}{\beta_n(V_{DD}-V_{Tn})} \\ &= \frac{1}{(150\times10^{-6})(6)(3.3-0.70)} \\ &= 427.35 \quad \Omega \end{aligned} \tag{7.60}$$

所以放电时间常数为

$$\tau_n = (427.35)(150\times10^{-15}) = 64.1 \quad \text{ps} \tag{7.61}$$

下降时间为

$$t_f = 2.2\tau_n = 141.0 \quad \text{ps} \tag{7.62}$$

综合这些结果，最大信号频率为

$$f_{max} = \frac{1}{t_r+t_f} = \frac{1}{(271.55+141.0)\times10^{-12}} = 2.42 \quad \text{GHz} \tag{7.63}$$

式中 1 GHz＝10^9 Hz。虽然这是一个非常高的频率，但一定要记住它只是指单个反相器。

7.2.3 传播延时

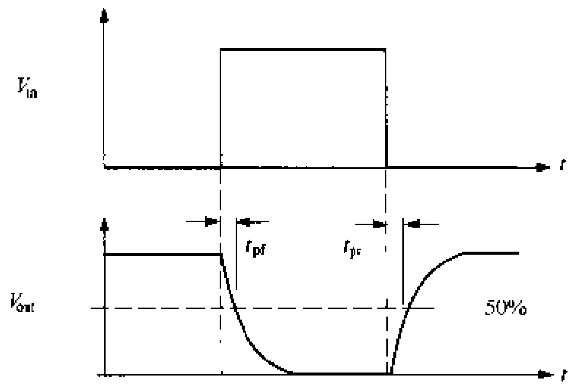


图 7.14 传播时间定义

传播延时时间 t_p 通常用来估计由输入到输出的“反应”延迟时间。当用阶跃输入电压时，传播延时定义为图 7.14 所示的两个时间间隔的简单平均数：

$$t_p = \frac{(t_{pf}+t_{pr})}{2} \tag{7.64}$$

式中，t_{pf}是输出由电源最高电平到“50%”，即从 V_{DD}到($V_{DD}/2$)的下降时间；t_{pr}是从 0 V 到($V_{DD}/2$)的传播上升时间。利用 V_{out}的指数方程得到

$$\begin{aligned} t_{pf} &= \ln(2)\tau_n \\ t_{pr} &= \ln(2)\tau_p \end{aligned} \tag{7.65}$$

近似 ln(2)≈0.693 得到

$$t_p \approx 0.35(\tau_n+\tau_p) \tag{7.66}$$

传播延时对估计基本延时非常有用,但是不能提供上升和下降时间各为多少的细节信息。传播延时普遍用于基本的逻辑模拟程序。

7.2.4 一般分析

上升和下降时间公式是高速 CMOS 设计的基础。可以通过它们了解如何设计单个的逻辑门,以及当用在逻辑链中时这些门的行为特性。

为了了解重要因素,回想一下总输出电容由两项构成:

$$C_{out} = C_{FET} + C_L \tag{7.67}$$

C_{FET}代表晶体管的寄生电容,C_L 是外部负载。C_{FET}值可由版图几何图确定,而负载电容 C_L 却随应用的不同而改变。将上式代入上升和下降时间公式,得到

$$\begin{aligned} t_r &\approx 2.2R_p(C_{FET} + C_L) \\ t_f &\approx 2.2R_n(C_{FET} + C_L) \end{aligned} \tag{7.68}$$

可以改写成以下形式

$$\begin{aligned} t_r &= t_{r0} + \alpha_p C_L \\ t_f &= t_{f0} + \alpha_n C_L \end{aligned} \tag{7.69}$$

上式表明上升和下降时间是负载电容 C_L 的线性函数。这两个量的一般特点见图 7.15。在零负载情况时($C_L=0$),反相器驱动其自身的电容,于是

$$\begin{aligned} t_r &= t_{r0} \approx 2.2R_pC_{FET} \\ t_f &= t_{f0} \approx 2.2R_nC_{FET} \end{aligned} \tag{7.70}$$

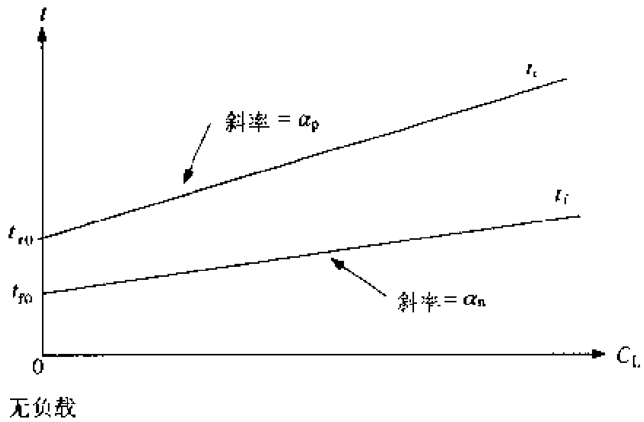


图 7.15 上升和下降时间的一般特点

它们完全由反相器参数决定。当加入一个外部负载 C_L 时,开关时间以线性方式增加。大的电容负载由于延迟时间较长可能引起问题。这一线性关系可用斜率值描述:

$$\alpha_p = 2.2R_p = \frac{2.2}{\beta_p(V_{DD} - |V_{Tp}|)} \tag{7.71}$$

和

$$\alpha_n = 2.2R_n = \frac{2.2}{\beta_n(V_{DD} - V_{Tn})} \tag{7.72}$$

注意它们与宽长比成反比，因为

$$\beta_p = k'_p\left(\frac{W}{L}\right)_p, \qquad \beta_n = k'_n\left(\frac{W}{L}\right)_n \tag{7.73}$$

对于一个给定的负载电容 C_L，可以采用大的FET来减小 t_r 和 t_f。然而，晶体管宽长比的增加意味着它将占用较大的芯片面积，从而减少了放在该电路芯片面积上的器件数目。因此，设计以提高速度为目标会降低电路的集成密度。这称为**速度与面积间的综合考虑**，即

快的电路比慢的电路消耗更多的面积

芯片设计者总是面临着减少开关延时而又不希望要求过多的芯片面积的问题，用行话来说，芯片面积又叫硅片"地产"。

【例7.4】 让我们用例7.3的结果，求FET内部电容为 $C_{FET}=80$ fF情况下的一般的延时公式。

上升时间 t_r 由pFET控制，它的电阻为 $R_p=822.9\ \Omega$，斜率为

$$\alpha_p = 2.2R_p = 1\ 810.4\ \Omega \tag{7.74}$$

而

$$\begin{aligned} t_{r0} &\approx 2.2R_pC_{FET} \\ &= 2.2(822.9)(80\times10^{-15}) \\ &= 144.9\ \text{ps} \end{aligned} \tag{7.75}$$

所以上升时间可写成下列形式

$$\begin{aligned} t_r &= t_{r0} + \alpha_pC_L \\ &= 144.9 + 1.810C_L\ \text{ps} \end{aligned} \tag{7.76}$$

它要求 C_L 的单位为fF。

为了求下降时间公式，我们计算

$$\alpha_n = 2.2(427.35) = 940.2\ \Omega \tag{7.77}$$

及

$$t_{f0} = 2.2(940.2)(80\times10^{-15}) = 165.5\ \text{ps} \tag{7.78}$$

得到

$$t_f = 165.5 + 0.940C_L\ \text{ps} \tag{7.79}$$

这就是延时的一般表达式。

作为使用这些公式的例子，假设负载定为 $C_L=150$ fF。计算输出端的上升和下降时间为：

$$\begin{aligned} t_r &= 144.9 + 1.810(150) = 416.4\ \text{ps} \\ t_f &= 165.5 + 0.940(150) = 306.5\ \text{ps} \end{aligned} \tag{7.80}$$

它对应的该门的最大开关频率 $f_{max}\approx 1.38$ GHz。

$(W/L)_n$ 和 $(W/L)_p$ 的相对值决定了输出波形的形状。例如，设计的电路有

$$R_p = R_n \quad (7.81)$$

则输出波形是对称的，即

$$t_r = t_f \quad (7.82)$$

要使电阻相等，设计的电路必须满足

$$\beta_p(V_{DD} - |V_{Tp}|) = \beta_n(V_{DD} - V_{Tn}) \quad (7.83)$$

如果 $V_{Tn} = |V_{Tp}|$，则只需要

$$\beta_p = \beta_n \quad (7.84)$$

这使 DC 的中点电压 $V_M = (V_{DD}/2)$。它说明了 nFET/pFET 之比 (β_n/β_p) 决定了 DC 的中点电压，而 β_n 和 β_p 各自的值分别决定了开关时间 t_f 和 t_r。

7.2.5 反相器电路小结

至此很值得用一点时间将学习小结一下。一个单独立的 CMOS 反相器的电气特性是由两组参数确定的：

- 工艺变量，例如 k' 和 V_T 值，以及寄生电容；
- 晶体管宽长比 $(W/L)_n$ 和 $(W/L)_p$。

VLSI 设计者无法控制工艺参数，因为它们是由生产流程细节确定的。所以，器件的尺寸成为高速电路设计的关键问题。

系统设计是通过把逻辑门链接起来，进行所需要的二进制运算完成的。用电学术语来说，逻辑信号流路径确定了每个门看到的负载电容 C_L。宽长比的选择是达到所希望的逻辑门链瞬态响应的关键。

7.3 功耗

CMOS 集成电路的一个重要特点就是一种具体设计总会有功耗。其一般问题显示于图 7.16 中。电流 I_{DD} 从电源流向地的功耗为

$$P = V_{DD} I_{DD} \quad (7.85)$$

由于把电源电压值 V_{DD} 假设为常数，可以通过研究电流的特性来求 P 值。一般把电流分成直流(DC)和动态电流(或开关电流)两部分，所以可以写成

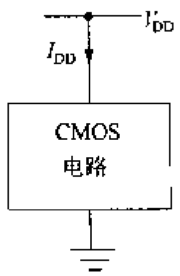


图 7.16 功耗计算的基础

$$P = P_{DC} + P_{dyn} \quad (7.86)$$

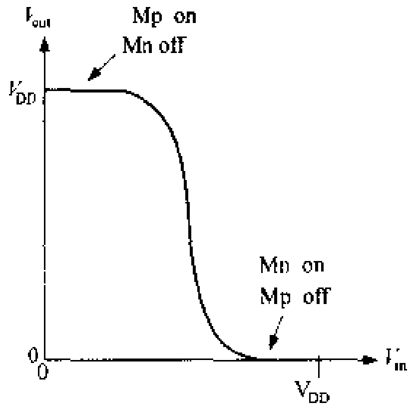
其中 P_{DC} 是 DC 功耗，P_{dyn} 是由于动态切换造成的功耗。

DC 部分功耗可以通过图 7.17(a) 重现的电压传输曲线来计算。当输入电压 V_{in} 稳定在低逻辑 0 值时，nFET Mn 截止；正如前面图 7.2 所示，在 V_{DD} 和地之间没有直接的电流通路。理

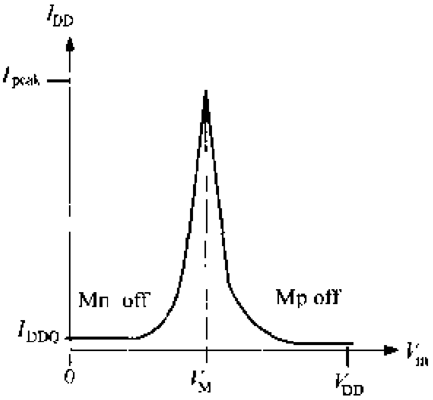
想情况下，DC 电流这时应为 $I_{DD}=0$，但在实际电路中，还存在很小的**漏电电流**。④ 其值用符号 I_{DDQ}表示，称为**静态漏电流**。当 V_{in}被切换，电流在 V_M 处达到峰值 I_{peak}时，如图 7.17(b)所示。当输入达到逻辑 1 的电压时，pFET Mp 截止，电流的直接通路再一次被切断。如果假设输入稳定在 0 或 1 状态为一个静止系统，DC 功耗则为

$$P_{DC} = V_{DD}I_{DDQ} \tag{7.87}$$

漏电流 I_{DDQ}通常很小，每个门的典型值在微微安培数量级上，因而 P_{DC}的值也很小。这个考虑是 20 世纪 90 年代中转向 CMOS 的主要因素。



(a) VTC



(b) DC 电流

图 7.17 DC 电流

为了求动态功耗 P_{dyn}，采用方波输入电压 $V_{in}(t)$，如图 7.18(a)所示。方波的周期为 T，对应的开关频率为

$$f = \frac{1}{T} \tag{7.88}$$

单位为赫兹；频率是在一秒钟内完成的周期数。在第 1 个半周期中，输入电压值为 $V_{in}=0$。这使 pFET Mp 导通，如图 7.18 (b)所示。由于 nFET 截止，电流 i_{DD}流过 Mp 并使 C_{out}充电至电压为 $V_{out}=V_{DD}$。在第 2 个半周期，输入电压为高，使 nFET Mn 导通。这引起放电如图 7.18(c) 所示，使 V_{out}衰减至 0 V。通过观察一个完整的周期电流是如何形成从电源到地的路径，就可理解动态功耗是怎样产生的：在充电过程中电流流向电容 C_{out}，加上该电容放电至地的路径就一起形成一个完整的电路。

为了计算 P_{dyn}，我们注意到充电过程使 C_{out}的电压成为 $V_{out}=V_{DD}$。这相当于电容上储存了电荷，其值为：

$$Q_e = C_{out}V_{DD} \tag{7.89}$$

单位为库仑。当该电容通过 nFET 放电时就会失去同样数量的电荷。在一个周期 T 内的平均功耗为

$$P_{av} = V_{DD}I_{DD} = V_{DD}\left(\frac{Q_e}{T}\right) \tag{7.90}$$

④ 将在第 9 章中详细讨论。

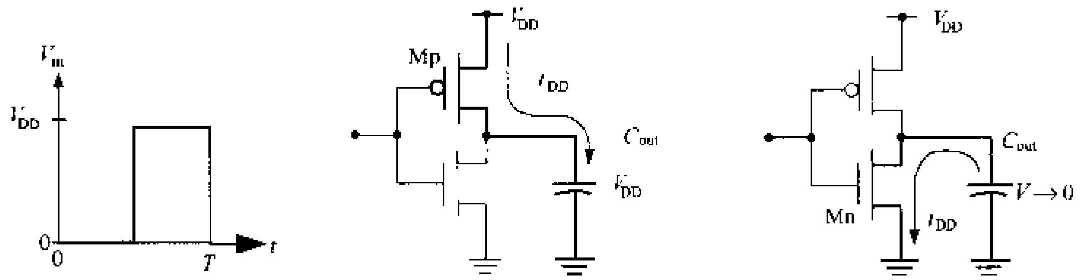
代人 Q_e 得到开关功耗为：

$$P_{sw} = C_{out} V_{DD}^2 f \qquad (7.91)$$

将 DC 和动态功耗两项合并得到总功耗为

$$P = V_{DD} I_{DDQ} + C_{out} V_{DD}^2 f \qquad (7.92)$$

上式中通常动态功耗一项占主要部分。这说明了一个极为重要的问题：动态功耗与信号频率成正比。也就是说，速度快的电路比速度慢的电路消耗更多的功耗。如果使开关速度加倍，则动态功耗也增加一倍。这些实际上就是物理定律的陈述，即必须提供能量才能引起电路中的变化。电路的切换不可能不消耗能量。



(a) 输入电压　　(b) 充电　　(c) 放电

图 7.18　求暂态功耗的电路

7.4　DC 特性：与非门（NAND 门）和或非门（NOR 门）

前面介绍的对于反相器电路的基本计算也可用来分析与非门（NAND 门）和或非门（NOR 门）。它们的 DC 和过渡特性都可以用较为简单的方法得到。在本节中，将考察器件尺寸与 VTC 描述的过渡特性之间的关系。

7.4.1　与非门（NAND）分析

让我们从图 7.19 的 NAND2 门开始，分析所有同极性 FET 具有同样宽长比的情况。这意味着两个 pFET 都用 β_p 描述，而两个 nFET 都有同样的 β_n。由于 pFET 是并联而 nFET 是串联，因此电路的特性与简单的反相器有显著差别。

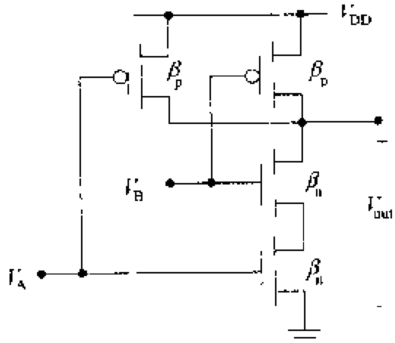
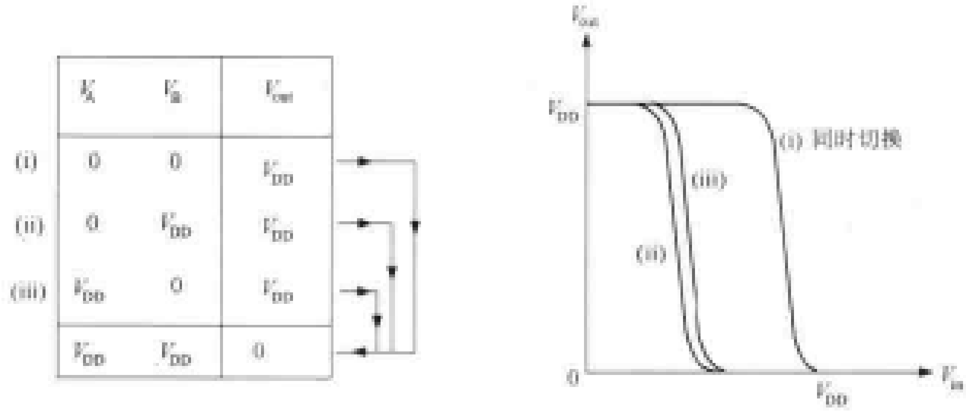


图 7.19　NAND2 逻辑电路

存在两个独立输出意味着需要有一个以上的 VTC 曲线来描述该电路。假设需要了解 V_{out} 起始为高电平 V_{DD}，随后在输入改变时下降到 0 V 的过渡情况。图 7.20(a) 概括了可以形成这种情况的可能的起始点。在情形(i)，V_A 和 V_B 均为 0 V，然后切换到最下面一行的情况，即 V_A =

$V_B = V_{DD}$，这时 $V_{out} = 0$ V。由于两个输入同时上升，它描述的是输入同时切换的情况。另外两种可能(ii)和(iii)描述的是只有一个输入改变的情形。例如在(ii)中，V_A 从 0 V 变化到 V_{DD}，而 V_B 则保持在 V_{DD} 不变。这三种可能性形成了图 7.20(b)上的三条不同的过渡曲线。它表明，两个输入同时切换的曲线较单个输入切换的曲线要"向右靠"。

利用版图计算同时切换情况下的中点电压值 V_M 很有指导意义。这一电路问题表示在图 7.21 中，其中 W_n 和 W_p 分别为 nFET 和 pFET 的沟道宽度。假设所有晶体管的沟道长度均为 L。那么在这种情况下输入电压 V_A 和 V_B 都等于 V_M。这样，在版图上两条栅都处于相同的电位，因此可以连在一起以简化计算。



(a) 过渡情况表　　　(b) TVC 曲线族

图 7.20　NAND2 VTC 分析

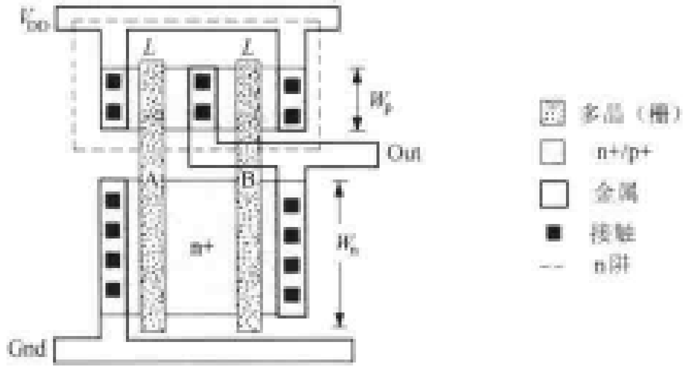
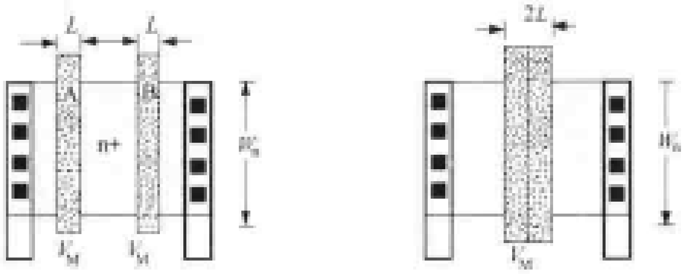


图 7.21　用于 V_M 计算的 NAND2 版图

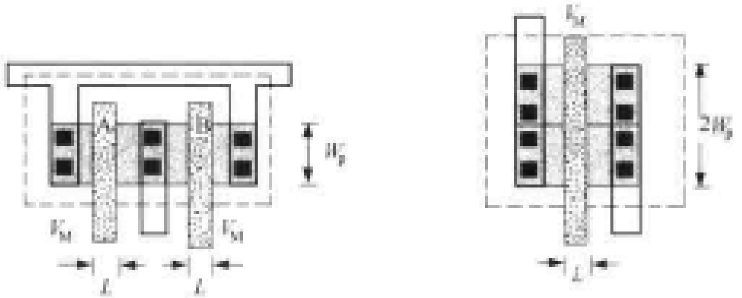
首先考虑 nFET。图 7.22(a)表现的是版图的原有样式，其中含有两个分开的串联的晶体管。让我们将这两条栅合二为一，得到图 7.22(b)的图样。如果忽略分隔两条栅的 n+ 区，则该结构可以近似为一个宽长比为($W_n/2L$)的 nFET。既然原来的每个 nFET 都有一个器件互导 β_n，那么这一单个等效的晶体管就可以用互导值($\beta_n/2$)来描述。

pFET 也可以用类似的方式来合并。图 7.23(a)表示原来并联的晶体管。由于是并联连接，左右两边在电气上是同一点，所以可以将其简化成一条栅结构，如图 7.23(b)所示。这时，两个晶体管合起来的作用如同一个宽长比为($2W_p/L$)的 pFET。如果原来每个器件有一个 β_p，那么这个等效结构的作用像一个具有 $2\beta_p$ 的 pFET。



(a) 两个单独的晶体管　　(b) 等效的单个 FET

图 7.22　串联 nFET 的简化



(a) 两个单独的晶体管　　(b) 等效的单个 FET

图 7.23　并联 pFET 的简化

让我们用这些结果来求输入同时切换时的 V_M。将这两对晶体管用它们等效的单个 FET 来替代，就得到了如图 7.24 所示的反相器电路，它的 nFET 和 pFET 的互导分别为 $(\beta_n/2)$ 和 $2\beta_p$。然后就可以用与"正常"非门同样的方式进行计算了。两个晶体管都是饱和的，所以使它们的电流相等得到

$$\frac{(\beta_n/2)}{2}(V_M - V_{Tn})^2 = \frac{(2\beta_p)}{2}(V_{DD} - V_M - |V_{Tp}|)^2 \qquad (7.93)$$

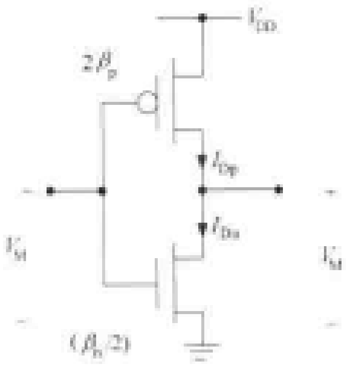


图 7.24　NAND2 门求 V_M 的简化电路

两边同取平方根并且求解中点电压值得到下列表达式

$$V_M = \frac{V_{DD} - |V_{Tp}| + \frac{1}{2}\sqrt{\frac{\beta_n}{\beta_p}} V_{Tn}}{1 + \frac{1}{2}\sqrt{\frac{\beta_n}{\beta_p}}} \tag{7.94}$$

它和非门公式(7.14)的形式相同，只是每个平方根项乘以一个因子(1/2)。这使分母变小，这就是为什么VTC曲线右移的原因。如果应用同样的道理于 N 个输入的与非门，就可以求得同时切换点为

$$V_M = \frac{V_{DD} - |V_{Tp}| + \frac{1}{N}\sqrt{\frac{\beta_n}{\beta_p}} V_{Tn}}{1 + \frac{1}{N}\sqrt{\frac{\beta_n}{\beta_p}}} \tag{7.95}$$

右移是由于串联的nFET造成的，因为它们的电阻相加。

7.4.2 或非门(NOR门)

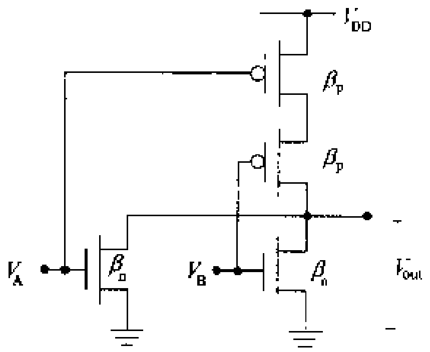
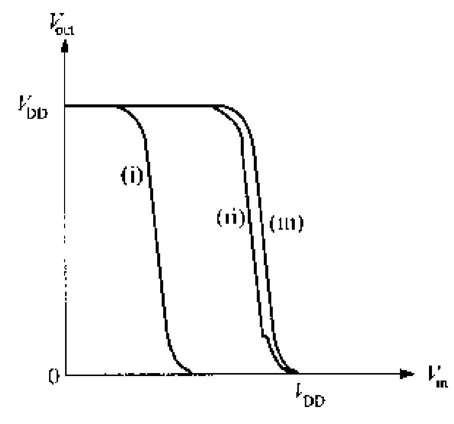


图7.25 NOR2电路

可用同样的方法来分析NOR2门。假设nFET具有同样的 β_n，并且两个pFET均用 β_p 来描述如图7.25所示的基本电路。要画出VTC，注意 $V_{out} = V_{DD}$ 要求 $V_A = V_B = 0$ V。如果任一输入(或两个同时)切换到逻辑1的值，则输出将下降至 $V_{out} = 0$ V。三种组合列于图7.26(a)的功能表中。与NAND2门一样，也有三条不同的过渡曲线，如图7.26(b)所示的VTC曲线族。情形(i)描述同时切换的情况，此时 V_A 和 V_B 都从0 V增加到 V_{DD}。这条曲线位于VTC曲线族的最左边，这恰与NAND2的情况相反。单个输入切换曲线(ii)和(iii)不同，但互相靠得很近。

	V_A	V_B	V_{out}
	0	0	V_{DD}
(iii)	0	V_{DD}	0
(ii)	V_{DD}	0	0
(i)	V_{DD}	V_{DD}	0

(a) 过渡情况表



(b) VTC曲线族

图7.26 NOR2三种过渡情况下的VTC

合并串、并联晶体管的方法可以用来计算同时切换情况下的 V_M 值。由于 nFET 是并联的，它们可以合并成互导为 $2\beta_n$ 的单个等效 nFET。而串联 pFET 的作用则像一个互导为 $(\beta_p/2)$ 的单个 pFET。由此得到图 7.27 所示的简化的等效电路。采用有效互导值且使饱和电流相等得到

$$\frac{(2\beta_n)}{2}(V_M - V_{Tn})^2 = \frac{(\beta_p/2)}{2}(V_{DD} - V_M - |V_{Tp}|)^2 \tag{7.96}$$

求解上式得到

$$V_M = \frac{V_{DD} - |V_{Tp}| + 2\sqrt{\frac{\beta_n}{\beta_p}}V_{Tn}}{1 + 2\sqrt{\frac{\beta_n}{\beta_p}}} \tag{7.97}$$

将其与“非门”和“与非门”公式比较可以看出，惟一的差别在于平方根项乘以 2 这个因子。这使分母变大，从而使 V_M 的值较器件比值为(β_n/β_p)的反相器要小。N 个输入的或非门的中点电压为

$$V_M = \frac{V_{DD} - |V_{Tp}| + N\sqrt{\frac{\beta_n}{\beta_p}}V_{Tn}}{1 + N\sqrt{\frac{\beta_n}{\beta_p}}} \tag{7.98}$$

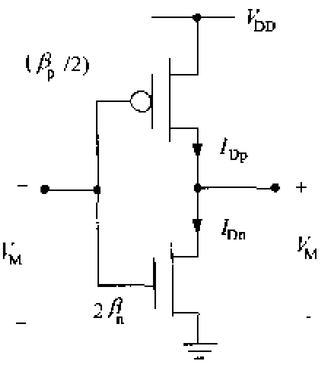


图 7.27　NOR2 门输入同时切换时 V_M 的计算

值得注意的是，与非门和或非门相对于参照非门的 VTC 而言，往往具有彼此相反的行为特点。

最后，我们注意到，与非门和或非门都表现出低的直流功耗值：

$$P_{DC} = V_{DD}I_{DDQ} \tag{7.99}$$

这是由于当输入稳定在逻辑 0 或逻辑 1 值时，从电源到地没有任何直接的电流通路。这些门的低功耗特点是由于使用了互补对和串-并联结构的晶体管阵列的缘故。动态功耗仍为以下一般形式：

$$P_{sw} = C_{out}V_{DD}^2 f_{gate} \tag{7.100}$$

由此可以看到它与门开关频率 f_{gate} 的关系。由于它使用一个以上的输入来切换门，f_{gate} 与反相器的基本开关频率不同。这将在以后做更详细的讨论。

7.5　与非门和或非门的暂态响应

过渡的开关时间经常是数字逻辑链设计中的限制因素。本节将考察 FET 拓扑连接和器件尺寸如何影响门的工作速度。

7.5.1 NAND2 开关时间

考虑图 7.28 中的 NAND2 门。其总输出电容表示为

$$C_{out} = C_{FET} + C_L \tag{7.101}$$

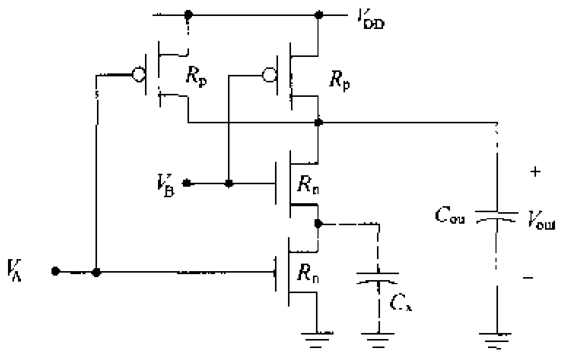


图 7.28 用于暂态计算的 NAND2 电路

其中 C_L 为外加负载，而

$$C_{FET} = C_{Dn} + 2C_{Dp} \tag{7.102}$$

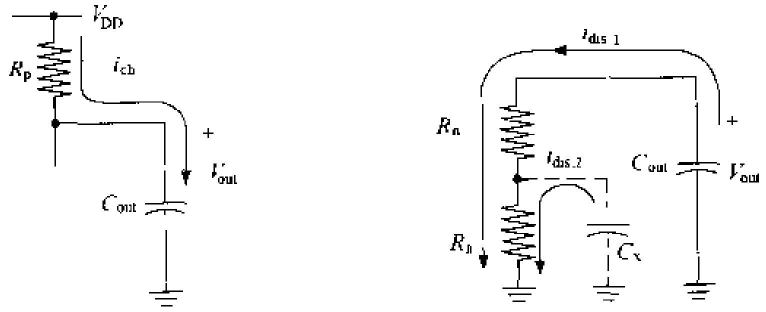
代表 FET 内部的寄生电容。注意，因为有两个 pFET 与输出节点相连，所以有两个 C_{Dp}。图中标出了晶体管的电阻值：

$$R_p = \frac{1}{\beta_p(V_{DD} - |V_{Tp}|)}, \qquad R_n = \frac{1}{\beta_n(V_{DD} - V_{Tn})} \tag{7.103}$$

暂态计算是以求晶体管的充电时间（t_r 或 t_{LH}）和放电（下降）时间（t_f 或 t_{HL}）的 RC 时间常数为基础的。这一过程因存在两个输入而较为复杂。我们将集中于估算最坏情况下的开关时间值。

首先考虑上升时间 t_r。输出电压的起始值为 $V_{out}(0) = 0$ V，然后充电到 V_{DD}。如果只有一个 pFET 导电，我们就得到如图 7.29(a)所示的简化充电电路，图中 C_{out} 通过一个 pFET 电阻 R_p 充电。这看上去像一个简单反相器的充电电路，可以写出

$$V_{out}(t) = V_{DD}[1 - e^{-t/\tau_p}] \tag{7.104}$$



(a) 充电电路 (b) 放电电路

图 7.29 估算上升和下降时间的 NAND2 的子电路

式中

$$\tau_p = R_p C_{out} \tag{7.105}$$

为时间常数。从而上升时间为

$$t_r \approx 2.2\tau_p \tag{7.106}$$

这可看成"最坏"情况,因为只有一个 pFET 在充电 C_{out}。注意它可以改写成线性形式:

$$t_r = t_0 + \alpha_0 C_L \tag{7.107}$$

式中

$$t_0 = 2.2 R_p C_{FET} \tag{7.108}$$

是零负载时的值,同时

$$\alpha_0 = 2.2 R_p \tag{7.109}$$

是 t_r 作为负载电容 C_L 函数的斜率。如果两个 pFET 都导电,那么等效电阻降为 $(R_p/2)$,因为二者是并联的;这将是最好情况,即它的充电时间最短。设计通常以最坏情况分析为基础,因为需要保证所有情况下电路都工作。

当要分析 C_{out} 通过串联 nFET 链放电的下降时间 t_f 时,情况变得更加复杂。每个器件的 RC 模型产生如图 7.29(b)所示的"梯形"网络。虽然我们主要关心的是 C_{out} 的放电,但由于两个 n 沟道晶体管间存在 FET 间的电容 C_X,情况变得复杂了。在最坏情况下,C_X 具有电荷且将通过 nFET MnA 流向地。由于通过一个 FET 的电流要受到其宽长比(W/L)的限制,放电速率自然也要受到 MnA 能够维持的电流的限制。

放电可以用输出电压指数形式的模型描述为

$$V_{out}(t) = V_{DD} e^{-t/\tau_n} \tag{7.110}$$

其时间常数是由 **Elmore 公式**得到

$$\tau_n = C_{out}(R_n + R_n) + C_X R_n \tag{7.111}$$

它将时间常数估计为各个时间常数的叠加

$$\tau_n = \tau_{n1} + \tau_{n2} \tag{7.112}$$

式中

$$\tau_{n1} = C_{out}(R_n + R_n) \tag{7.113}$$

是 C_{out} 通过两个电阻均为 R_n 的 nFET 放电的时间常数;这可以从图中的电流 $i_{dis,1}$ 看出。另一项

$$\tau_{n2} = C_X R_n \tag{7.114}$$

是 C_X 通过电阻为 R_n 的一个 nFET 放电的时间常数。它对应于放电电流 $i_{dis,2}$。于是下降时间 t_f 为

$$t_f \approx 2.2\tau_n \tag{7.115}$$

代入时间常数表达式后上式变为

$$t_f \approx 2.2[(C_{FET} + C_L)(2R_n) + C_X R_n] \tag{7.116}$$

合并各项得到线性表达式

$$t_f = t_1 + \alpha_1 C_L \tag{7.117}$$

式中 t_1 为零负载时的延时

$$t_1 = 2.2R_n(2C_{FET} + C_X) \tag{7.118}$$

而斜率为

$$\alpha_1 = 4.4R_n \tag{7.119}$$

上式中的系数来自于(2×2.2)。虽然可把 t_f 写成 C_L 的线性函数,但零负载的延时和斜率都受到放电电路中串联的 nFET 的影响。

RC 梯形网络时间常数的 Elmore 公式,表明在 CMOS 电路中串联 FET 会导致较长的延时。要理解这一点,可把式(7.111)重写为

$$\tau_n = R_n(2C_{out} + C_X) \tag{7.120}$$

在这一形式中,可把时间常数看成 R_n 与一个有效电容的乘积,该电容的值为

$$C_{eff} = 2C_{out} + C_X \tag{7.121}$$

它比 2 倍的输出电容大。另一种写法是:

$$\tau_n = C_{out}(2R_n) + C_X R_n \tag{7.122}$$

它清楚地表明串联 FET 在 $2R_n$ 一项中的影响,以及由于寄生电容 C_X 造成的延时增加。无论采用哪种解释,重要的是要记住串联 FET 链会导致过多的逻辑延迟。

7.5.2 二输入或非门(NOR2)的开关时间

NOR2 的过渡分析方式与上面相同。图 7.30 是包括 FET 电阻和电容的电路。任一个门的输出电容都可表示成下列一般形式:

$$C_{out} = C_{FET} + C_L \tag{7.123}$$

对于 NOR2 电路来说,因为有两个 nFET 但只有一个 pFET 与输出节点相连,所以内部电容可分解为两部分:

$$C_{FET} = 2C_{Dn} + C_{Dp} \tag{7.124}$$

FET 间的电容 C_y 代表在两个 pFET 之间的寄生电容。

图 7.31 是输出过渡情况下的子电路。下降时间 t_f 用图 7.31(a)的最坏情况电路来计算。在该电路中只有一个 nFET 对输出电容放电起作用,输出电压写为

$$V_{out}(t) = V_{DD}e^{-t/\tau_n} \tag{7.125}$$

式中

$$\tau_n = R_n C_{out} \tag{7.126}$$

为时间常数。于是下降时间为

$$t_f \approx 2.2\tau_n \tag{7.127}$$

这与简单反相器的情况相同。展开 C_{out} 得到线性关系

$$t_f = t_1 + \alpha_1 C_L \tag{7.128}$$

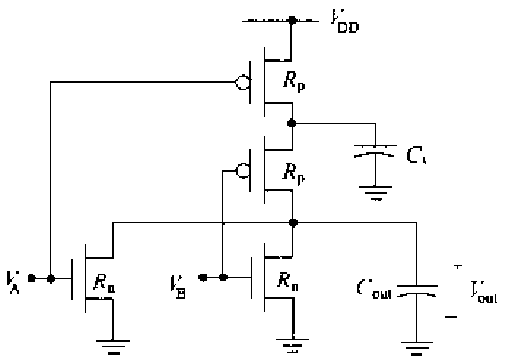
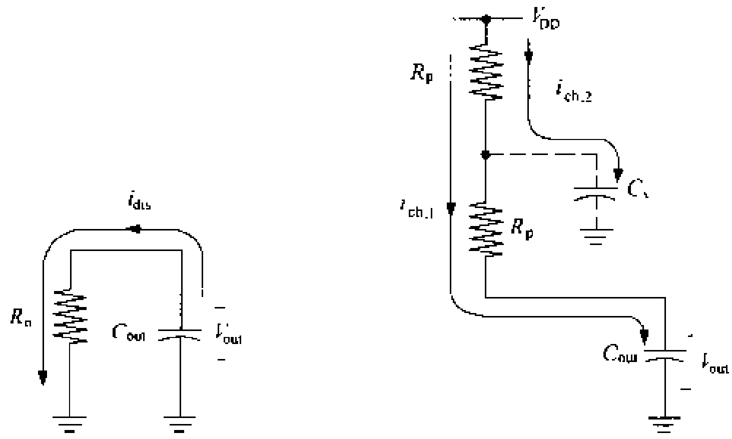


图 7.30　用于计算开关时间的 NOR2 电路



(a) 放电电路　　(b) 充电电路

图 7.31　用于 NOR2 暂态计算的子电路

式中零负载延时为

$$t_1 = 2.2 R_n C_{FET} \tag{7.129}$$

而斜率为

$$\alpha_1 = 2.2 R_n \tag{7.130}$$

这些结果与非门类似，但一定要记住 NOR2 门的 C_{FET}较大。

图 7.31(b)是求上升时间 t_r 的充电电路。输出电压公式可写成下列指数形式：

$$V_{out}(t) = V_{DD}[1 - e^{-t/\tau_p}] \tag{7.131}$$

由于在这一过程中 C_y 将被充电，须用 Elmore 公式来求时间常数。图中的两条路径用 $i_{ch,1}$和 $i_{ch,2}$来表示。主充电路径 $i_{ch,1}$的时间常数为

$$\tau_1 = C_{out}(R_p + R_p) \tag{7.132}$$

而与 $i_{ch,2}$有关的时间常数为

$$\tau_2 = C_y R_p \tag{7.133}$$

通过叠加得到总有效时间常数为

$$\begin{aligned} \tau_p &= \tau_1 + \tau_2 \\ &= C_{out}(2R_p) + C_y R_p \end{aligned} \tag{7.134}$$

所以上升时间为

$$t_r = 2.2\tau_p \tag{7.135}$$

由于串联 pFET 引起的时间常数较大，上升时间与下降时间相比可以非常大。替换 C_{out}得到线性方程

$$t_r = t_0 + \alpha_0 C_L \tag{7.136}$$

式中

$$t_0 = 2.2R_p(2C_{FET} + C_y) \tag{7.137}$$

及

$$\alpha_0 = 4.4R_p \tag{7.138}$$

这说明了 t_r 与 C_L 相关的特点。与 NAND2 门一样,串联 FET 的存在将会减慢相应的开关时间。

7.5.3 小结

上述分析表明与非门和或非门无论在 DC 还是在过渡特性上都表现出互补的特点,这是由于它们是用互补的串并联晶体管结构构成的。

虽则 DC 特性很重要,但主要的设计力量还是直接用在减少通过逻辑链的延时。通过前面的学习可以总结几条关于与非门和或非门与较简单的非门相比时所具有的一般规律。首先,上升时间可以写为下列形式:

$$t_r = t_0 + \alpha_0 C_L \tag{7.139}$$

而下降时间也具有相同的结构:

$$t_f = t_1 + \alpha_1 C_L \tag{7.140}$$

常数(上升时间公式中的 t_0 和 α_0,下降时间公式中的 t_1 和 α_1)取决于晶体管的寄生电阻和电容。非门的这些常数最小,所以常用它来作为参照。这当然是因为反相器仅由两个 FET 组成的缘故。一般来说,增加互补对晶体管会增加延迟时间,因为 C_{FET}增加了。一个逻辑门输入的数目称为**扇入**(FI)。由于每个输入都连至一对互补管,可以断定

- 开关延时随扇入的增加而增加

这就是说,如果二者都用同样尺寸的晶体管的话,一个 NAND3 门将比 NAND2 门要慢。当然,实际延时取决于负载电容 C_L 的值,即

- 开关延迟随外加负载的增加而增加

由于逻辑功能是用门的串接完成的,这一相关性的影响随电路的不同而不同。

让我们来概括一下与非门和或非门分析的结果。和反相器一样,这些门的电气特性取决于

- 工艺变量;
- 每个 FET 的宽长比$(W/L)_n$和$(W/L)_p$;

再有,串联晶体管引起两个器件之间寄生电容的问题。这一因素可以再总结一条:

- 版图几何图的细节影响逻辑门的暂态响应。

从而可以得出结论:电路的物理版图和结构是设计高速逻辑电路的一个关键因素。

7.6 复合逻辑门的分析

上述与非门和非门电路的分析方法也可以扩展,分析具有 AOI 和 OAI 结构的复合 CMOS 逻辑门。最重要的问题是与串联 FET 相关的过渡延时问题。

考虑图 7.32 中的复合逻辑门。它的串并联 FET 阵列执行以下逻辑功能:

$$f = \overline{x \cdot (y+z)} \tag{7.141}$$

图中的宽长比值是影响上升和下降时间的关键参数。下降时间取决于 nFET。如果假定它们都具有同样的尺寸：

$$\left(\frac{W}{L}\right)_{nx} = \left(\frac{W}{L}\right)_{ny} = \left(\frac{W}{L}\right)_{nz} \tag{7.142}$$

则可以用 nFET 电阻 R_n 来描述它们的每一个，最坏情况的下降时间发生在当 $x=1$，且“或”的输入 y 或 z 中只有一个为 1 的时候。其结果形成两个 FET 串联，它们必须完成输出电容的放电

$$C_{out} = C_{FET} + C_L \tag{7.143}$$

若串联链中的电容为 C_n，则时间常数为

$$\tau_n = R_n C_n + 2R_n C_{out} \tag{7.144}$$

由此得到下降时间为

$$\begin{aligned} t_f &= 2.2\tau_n \\ &= 2.2R_n[C_n + 2(C_{FET} + C_L)] \\ &= t_1 + \alpha_1 C_L \end{aligned} \tag{7.145}$$

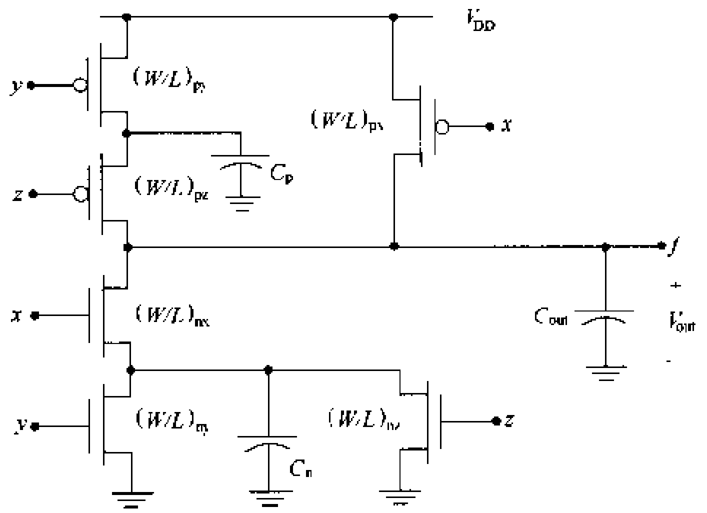


图 7.32　复合逻辑门电路

式中

$$t_1 = 2.2R_n(C_n + 2C_{FET}) \tag{7.146}$$

为零负载的时间，而

$$\alpha_1 = 2.2R_n \tag{7.147}$$

为斜率。

上升时间由 pFET 决定。如果它们都选用同样的宽长比

$$\left(\frac{W}{L}\right)_{px} = \left(\frac{W}{L}\right)_{py} = \left(\frac{W}{L}\right)_{pz} \tag{7.148}$$

那么每个器件都可以用同样的 R_p。限制上升时间的串联链是输入为 y 和 z 的 p 管；输入为 x

的 pFET 能够提供较快的切换，因而缩小到一半尺寸而不会影响结果。串联链的时间常数为

$$\tau_p = R_p C_p + 2R_p C_{out} \tag{7.149}$$

式中 C_p 是两个 pFET 之间的寄生电容。于是最坏情形的上升时间为

$$t_r = t_0 + \alpha_0 C_L \tag{7.150}$$

式中零负载延时为

$$t_0 = 2.2R_p(C_p + 2C_{FET}) \tag{7.151}$$

斜率为

$$\alpha_0 = 2.2R_p \tag{7.152}$$

一个任意的门产生同样形式的上升和下降时间公式，这说明这一推导过程的一般性。

一些重要的步骤很容易遵循。求最坏情形时的下降时间先要找到最长的 nFET 串联链。最长上升时间取决于最长串联的 pFET 链。两种情况都要用 Elmore 公式来计算时间常数，然后把各项分开求出零负载的延时和斜率。

功耗

回想一下简单反相器的功耗可写成如下形式：

$$P = V_{DD}I_{DDQ} + C_{out}V_{DD}^2 f \tag{7.153}$$

分析一个一般的静态 CMOS 逻辑门时，DC 项仍然很小，但动态开关功耗 P_{dyn} 在高速高密度设计中变得非常重要。

为了模拟一个任意门的动态功耗，可以回想一下 P_{dyn} 源自输出发生切换的过程。首先，输出电容 C_{out} 从 0 V 充电到 V_{DD}，相应于输出逻辑 0→1 的改变。然后，C_{out} 放电，产生 1→0 的转换，完成一个循环。为了模拟在一个开关周期 T 内发生转换的次数，引入**翻转系数** a 来代表在一个周期内发生一次输出 0→1 转换的概率。这样，动态功率改写为

$$P_{dyn} = aC_{out}V_{DD}^2 f \tag{7.154}$$

对于一个由 N 个门组成的电路，其总动态功率可以更为一般地写为

$$P_{dyn} = \sum_{i=1}^{N} a_i C_i V_i V_{DD} f \tag{7.155}$$

A	B	$\overline{A+B}$	$\overline{A\cdot B}$
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

图 7.33 确定翻转系数的真值表

式中，对于第 i 个门来说，a_i 是它的翻转系数，C_i 为充电到最大值 V_i 的节点电容。

翻转系数可从真值表确定。图 7.33 提供了 NOR2 和 NAND2 功能的真值表。将假定每个输入组合发生的可能性相同。首先分析 NOR2 的转换。由于翻转系数 a_{NOR2} 是门由 0→1 转换的概率，它可以计算为

$$a = p_0 p_1 \tag{7.156}$$

式中 p_0 是输出初始为 0 的概率，而 p_1 是它转换为 1 的概率。真值表显示 $p_0=(3/4)$，$p_1=(1/4)$，所以

$$a_{NOR2}=\left(\frac{3}{4}\right)\left(\frac{1}{4}\right)=\frac{3}{16} \tag{7.157}$$

NAND2 门可以用同样的方式分析。对于这个门，真值表显示 $p_0=(1/4)$，$p_1=(3/4)$，所以

$$a_{NAND2}=\left(\frac{3}{4}\right)\left(\frac{1}{4}\right)=\frac{3}{16} \tag{7.158}$$

它与 NOR2 门具有相同的值。如果看一下 3 输入的门，真值表给出

$$a_{NOR3}=\frac{7}{64}=a_{NAND3} \tag{7.159}$$

类似地，由于 $p_0=(1/4)=p_1$，我们可以计算出

$$a_{XNOR2}=\frac{1}{4}=a_{XOR2} \tag{7.160}$$

这一方法适用于任意一个门。

这个简单处理方法的局限性在于，在实际中很少有输入组合发生的概率相等的情况。已经开发更为先进的方法来处理这些情形，有兴趣的读者可以参阅参考资料[2]中对这些细节的绝妙讨论。参考资料[8]对功耗与低功耗设计做了全面的分析。

7.7 逻辑门过渡特性设计

高速电路受到各个门开关时间的限制。逻辑的构成决定晶体管的串并联连接。宽长比对 DC 和过渡开关时间来说都是关键的设计参数。这些一旦确定且晶体管已画成版图之后，所有的寄生参数也就确定了。

DC 的开关特性常被认为不如它的开关速度重要。通常设计一个门使它具有所期望的过渡时间，然后检查 DC 的 VTC 以保证它是可接受的。这一方法的基础是每个 nFET 和 pFET 的宽长比决定了开关响应，而 DC 切换点是 nFET 值与 pFET 值之比的结果。例如，一个反相器的 β_n/β_p 值决定了 V_M。而 t_r 主要取决于 β_p，t_f 取决于 β_n。

选择宽长比的设计原则要根据情况决定。最直接的办法是用反相器作为参照，然后努力设计其他门，使它们具有近似相等的开关时间。因为非门最为简单，所以可用较小的晶体管来构成。我们将用器件的互导

$$\beta=k'\left(\frac{W}{L}\right) \tag{7.161}$$

来代替宽长比。

图 7.34(a)是器件尺寸由 β_p 和 β_n 说明的反相器。这些值确定了电路的上升和下降时间 t_r 和 t_f，它用来作为参照的开关时间。由于两个晶体管驱动相同的电容，它们的差别在于电阻值

$$R_p=\frac{1}{\beta_p(V_{DD}-|V_{Tp}|)},\qquad R_n=\frac{1}{\beta_n(V_{DD}-V_{Tn})} \tag{7.162}$$

前面已讲到一个对称反相器具有

$$\beta_n = \beta_p \tag{7.163}$$

且要求器件尺寸关系为

$$\left(\frac{W}{L}\right)_p = r\left(\frac{W}{L}\right)_n \tag{7.164}$$

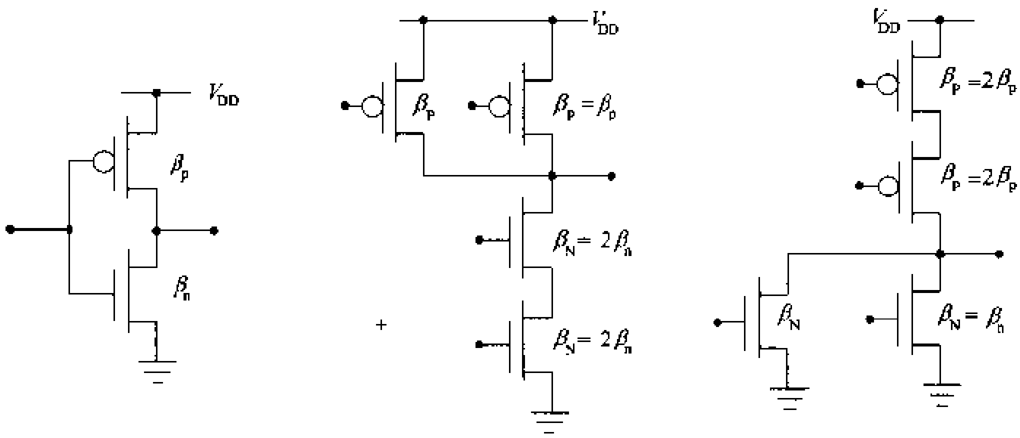
式中

$$r = \frac{k'_n}{k'_p} \tag{7.165}$$

为工艺互导比。使用相同尺寸晶体管，$\beta_n > \beta_p$ 的非对称设计也常常用来作为参照。

让我们用这些值来求图 7.34(b)中 NAND2 门的器件尺寸 β_P 和 β_N，原则是得到类似的上升和下降时间。首先考虑并联 pFET。由于最坏情况是只有一个晶体管对上升时间起作用，可以选择与反相器同样的尺寸：

$$\beta_P = \beta_p \tag{7.166}$$



(a) 反相器　　(b) NAND2　　(c) NOR2

图 7.34　FET 的相对尺寸

实际的上升时间 t 将比反相器的要长，因为 C_{out} 较大。串联的 nFET 链应被模拟为在输出与地之间的两个串联的电阻，总电阻为

$$R = R_N + R_N \tag{7.167}$$

式中

$$R_N = \frac{1}{\beta_N(V_{DD} - V_{Tn})} \tag{7.168}$$

用反相器作为参照，设

$$R = R_n = 2R_N \tag{7.169}$$

代入得到

$$\frac{1}{\beta_n(V_{DD} - V_{Tn})} = \frac{2}{\beta_N(V_{DD} - V_{Tn})} \tag{7.170}$$

由上式得到

$$\beta_N = 2\beta_n \tag{7.171}$$

即串联 nFET 的大小是反相器晶体管的 2 倍：

$$\left(\frac{W}{L}\right)_N = 2\left(\frac{W}{L}\right)_n \tag{7.172}$$

所得到的下降时间 t_f 将比 NAND2 门大，因为 NAND2 有较大的输出电容以及存在 FET－FET 间的内部电容。无论如何，这给了我们一个结构化的方法来确定门的尺寸。

图 7.34(c)中的 NOR2 门也可用同样的方式来设计。并联 nFET 选择与反相器器件具有相同的尺寸，因为这是最坏的放电情况，所以

$$\beta_N = \beta_n \tag{7.173}$$

串联 pFET 的电阻相加得到总电阻 $2R_P$。使它等于反相器电阻 R_p，得到

$$\frac{1}{\beta_P(V_{DD}-|V_{Tp}|)} = \frac{2}{\beta_P(V_{DD}-|V_{Tp}|)} \tag{7.174}$$

所以

$$\beta_P = 2\beta_p \tag{7.175}$$

这表明 pFET 是反相器晶体管大小的 2 倍：

$$\left(\frac{W}{L}\right)_P = 2\left(\frac{W}{L}\right)_p \tag{7.176}$$

主要问题在于 pFET 本质上较慢，所以 $(W/L)_p$ 可以在开始时选得较大。

这一方法可以扩展到更大的链中。对于 n 个串联的 FET，其尺寸必须 n 倍于反相器值。因此图 7.35(a)中的 NAND3 门可设计为

$$\beta_N = 3\beta_n, \qquad \beta_P = \beta_p \tag{7.177}$$

即有

$$\left(\frac{W}{L}\right)_N = 3\left(\frac{W}{L}\right)_n, \qquad \left(\frac{W}{L}\right)_P = \left(\frac{W}{L}\right)_p \tag{7.178}$$

而图 7.35(b)中的 NOR3 门则有

$$\beta_N = \beta_n, \qquad \beta_P = 3\beta_p \tag{7.179}$$

$$\left(\frac{W}{L}\right)_N = \left(\frac{W}{L}\right)_n, \qquad \left(\frac{W}{L}\right)_P = 3\left(\frac{W}{L}\right)_p \tag{7.180}$$

既然参考值 β_n 和 β_p 是任意的，晶体管的尺寸可按需要调整，得到合理的值。同时还要注意，如果选择 $\beta_n = \beta_p$ 的对称反相器设计，则所得到的门也将近似对称。

复合逻辑门也可以用同样的方式设计。考虑图 7.36 中的逻辑门，它采用串并联结构，输出为

$$f = \overline{(a \cdot b + c \cdot d) \cdot x} \tag{7.181}$$

首先考虑 nFET 阵列。任何放电过程都会有电流通过至少三个串联的 nFET。器件尺寸值可以一样，为

$$\beta_N = 3\beta_n = \beta_{N1} \tag{7.182}$$

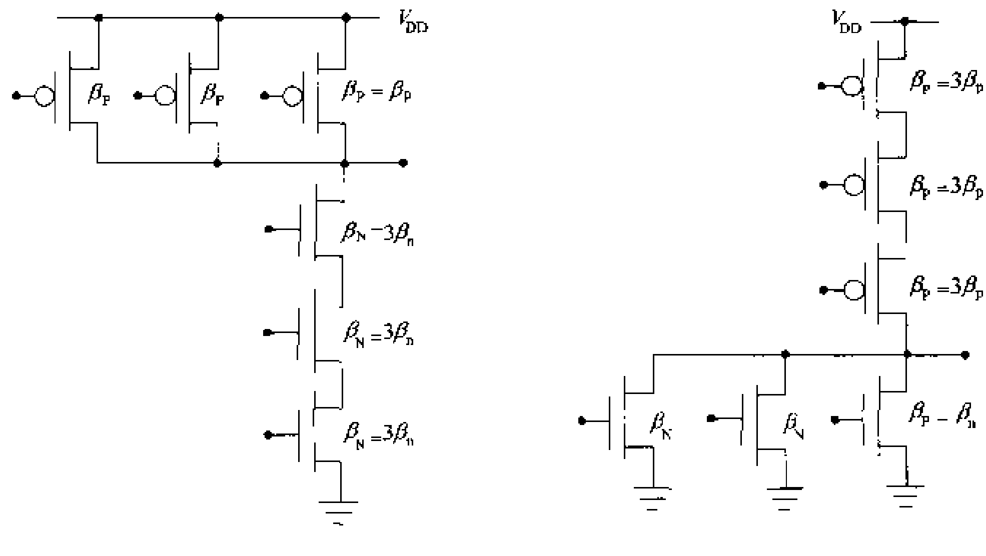


图 7.35　3 输入门尺寸的确定

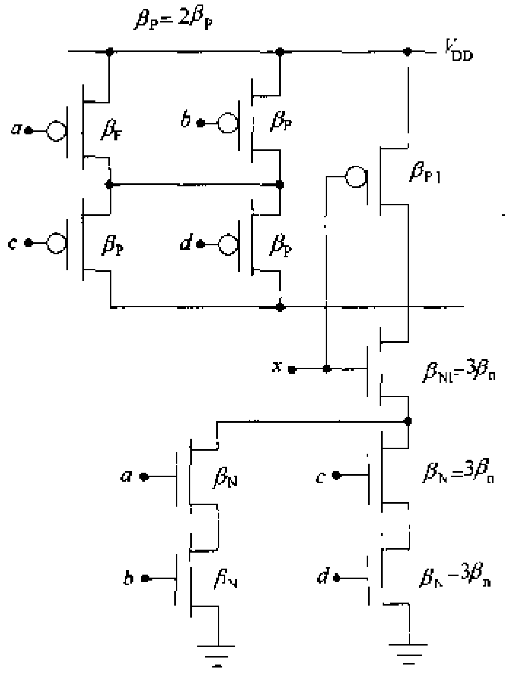


图 7.36　复合逻辑门尺寸的确定

pFET 阵列稍有不同。最坏情形的充电路径通过电路左边两个串联的晶体管。对于输入为 a，b，c 和 d 的 pFET,尺寸可以为

$$\beta_P = 2\beta_p \qquad (7.183)$$

输入为 x 的 pFET 是单独的，所以可将它的尺寸选择为与一个反相器一样：

$$\beta_{P1} = \beta_p \qquad (7.184)$$

但若改成另一种选择，即选输入为 x 的 pFET 为

$$\beta_{P1} = \beta_P = 2\beta_p \qquad (7.185)$$

则可以使版图比较简单，因为只使用一种尺寸的 pFET。

虽然这一方法提供了一个很好的结构化方法，但它却使晶体管较大。设计者必须决定为了提高速度而消耗芯片面积是否值得。当 FET 数目增加时这将变得更为复杂，因为在 Elmore 时间常数公式中 FET 到 FET 的各项寄生电容也将增大。在实际中，可先选择一个符合面积要求的标准单元，然后求出逻辑链的总速度。如果设计不够快，可以运用下一章中的某些技术来得到一个较好的设计。

7.8　传输门和传输管

传输门由并联在一起的一对 nFET/pFET 构成，如图 7.37(a)所示。图 7.37(b)所示的

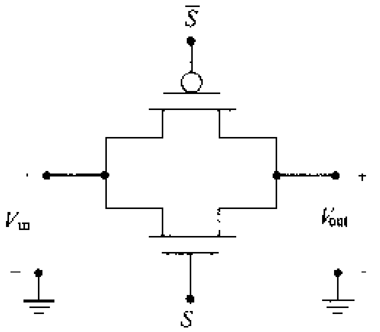
RC 开关模型由一个 TG 电阻 R_{TG}和考虑两个 FET 寄生参数的电容组成。虽然两个 FET 并联,但在任何一个给定时间通常只有一个对导电过程起着支配作用。例如,逻辑 0 的传递由 nFET 控制。由于这一点,对线性电阻的合理近似为

$$R_{TG} = \max(R_n, R_p) \tag{7.186}$$

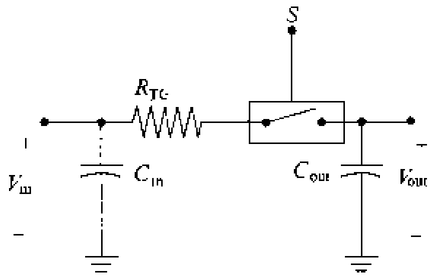
即采用两个值中较大的一个。电容由各部分相加得到。例如,假设左边比右边的电压低,那么

$$C_{in} = C_{S,n} + C_{D,p} \tag{7.187}$$

因为 nFET 的左边是源,而这个节点又是 pFET 的漏。[⑤]我们注意到,在选择这两个晶体管宽长比时要权衡利弊:(W/L)值大会降低电阻,但 W 值大又意味着电容大。这使得 TG 在高密度 VLSI 的发展中越来越缺乏吸引力。



(a) 电路



(b) RC 模型

图 7.37　传输门模型

传输门(和下面将讨论的传输管)的一个重要电气特性是信号没有直接连到电源 V_{DD}或地。静态逻辑电路由于有电源轨线可以提供全轨输出 $V_{OH}=V_{DD}$和 $V_{OL}=0$ V。由于 TG 没有采用这一方式,所以驱动电路(传输门前面的电路)起着提供输入信号电压的作用。然而 TG 成为该驱动门的 RC 寄生参数,所以其响应比没有 TG 存在时要慢。因此,需要加入另外的缓冲电路来保持速度。

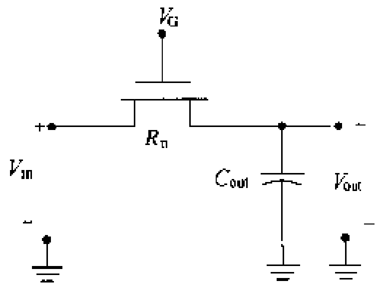


图 7.38　nFET 传输管

传输晶体管是单个的 FET,它在漏与源端之间传送信号而不是传送固定的电源值。在大多数电路中,"传输 FET"可以用来替代传输门。它们占用较少的面积及导线,但是不能通过全程电压。在对两种极性的管子做出选择时,常会选择 nFET,因为电子迁移率较大意味着 nFET 的开关速度比同样尺寸的 pFET 要快。

图 7.38 是基本 nFET 传输电路。开关由栅电压 V_G控制。如果 $V_G=0$,则晶体管截止,输入输出间没有联系。加上一个高电压 $V_G=V_{DD}$驱使 nFET 工作,使电流能够流动。对于传送逻辑 1,用一个输入电压 $V_{in}=V_{DD}$。假设起始条件为 $V_{out}(t=0)=0$,由分析得到[⑥]

⑤ 记住漏和源是由相对电压决定的。

⑥ 详细推导见参考资料[10]。

$$V_{out}(t) = V_{max}\left(\frac{t/2\tau_n}{1+t/2\tau_n}\right) \tag{7.188}$$

式中

$$V_{max} = V_{DD} - V_{Tn} \tag{7.189}$$

通过求极限

$$\lim_{t\to\infty} V_{out}(t) = V_{max} \tag{7.190}$$

可以看到式(7.189)是通过一个 nFET 的最大电压。它清楚地表明了阈值损失的问题。时间常数定义为

$$\tau_n = R_n C_{out} \tag{7.191}$$

但是它与出现在指数项中的时间常数含义不同。使输出电压从 0 V 上升到 $0.9V_{max}$所需要的上升时间可以计算为:

$$t_r = 18\tau_n \tag{7.192}$$

这些结果说明,传送逻辑 1 较慢且存在阈值损失问题。

逻辑 0 的传送可以通过使 $V_{in}=0\,V$ 来分析。在初始条件为 $V_{out}(0)=V_{max}$时,由分析得到

$$V_{out}(t) = V_{max}\left(\frac{2e^{-(t/\tau_n)}}{1+e^{-(t/\tau_n)}}\right) \tag{7.193}$$

式中时间常数具有同样的定义。该指数函数的极限为

$$\lim_{t\to\infty} V_{out}(t) = 0 \tag{7.194}$$

它说明一个 nFET 可以传送逻辑 0 而不会有任何问题。使输出从 V_{max}改变到 10% 电压即 $0.1V_{max}$时所需要的下降时间为

$$t_f = \ln(19)\tau_n \approx 2.94\tau_n \tag{7.195}$$

比较上升和下降时间可知

$$t_r \approx 6t_f \tag{7.196}$$

所以上升时间是限制因素。图 7.39 所示的曲线是一个 nFET 传输晶体管输出比照输入波形的例子。

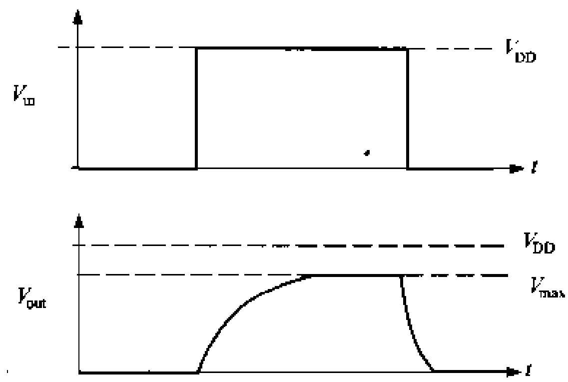


图 7.39 nFET 传输管的电压波形

若用 pFET 做传输管,可以得到互补的结果。通过 FET 的最大电压是 V_{DD},并且输出充电非常快,上升时间为

$$t_r = 2.94\tau_p \tag{7.197}$$

式中

$$\tau_p = R_p C_{out} \tag{7.198}$$

所以 pFET 能够传送一个强逻辑 1 电压。但是当在输入端加上逻辑 0 时,输出放电至

$$V_{min} = |V_{Tp}| \tag{7.199}$$

这时下降时间为

$$t_f = 18\tau_p \tag{7.200}$$

因此放电是限制因素。这些结果正是我们所预期的,因为 pFET 和 nFET 的行为互补。

分析表明传输晶体管不能精确地模拟为简单的 RC 电路,因为这样阈值损失和非对称的上升和下降时间会被忽略。尽管如此,在初始设计阶段的手工计算中,还是常常用 R_n 和 R_p 来模拟传输 FET。这使我们能够较快地模拟估算,因而是一个有价值的近似方法。更为精确的计算可以通过计算机模拟来完成。

7.9 关于 SPICE 模拟

本章进行的分析为设计 CMOS 逻辑门提供了理论基础。它能估计一个电路的行为,并且说明总的性能与各个器件参数间的依存关系。

解析分析受器件模型精度的固有限制。在 MOSFET 的情况下,平方律模型只是对电路真实行为的低阶近似。我们介绍的另一层估计假设了阶跃的输入电压。为了简化分析,还忽略了电容对于电压的相关性。在芯片设计中,一个电路的工作必须经过计算机模拟验证。这些并不容易,因为收敛问题和计算噪声都可以影响结果。一旦设计者熟悉了这些问题的领域,它们确实能够提供合理的验证。在本节中,将考察 SPICE 模拟的一些重要特点。

一个电路的 SPICE 网表来自版图编辑器的提取程序。每个元件由网表中单独的一行来表示,元件根据版图来连接。为了进行模拟,必须加上电源值、输入电压和模型信息。举例来讲,假设从一个反相器的版图中提取网表,得到以下各行:

```
M1 15 17 20 20 NFET W=5U L=0.5U
M2 15 17 12 12 PFET W=10U L=0.5U
```

它用随意的器件和节点编号定义了两个晶体管。在上列两行中,M1 是一个 nFET 而 M2 是一个 pFET。由于 MOSFET 节点的顺序是漏-栅-源-衬底,所以反相器的输入是公共的栅节点 17,输出则从漏节点 15 取出。节点 20 必定接地,而节点 12 则接电源。有些功能更强的提取程序还提供漏和源的尺寸,用来计算结电容,其形式如下:

```
M1 15 17 20 20 NFET W=5U L=0.5U AD=12.5P PD=15U AS=20P PS=18U
M2 15 17 12 12 PFET W=10U L=0.5U AD=25P PD=25U AS=40P PS=36U
```

如果提取程序未找到漏和源的面积和周长,那么必须人工加上。

若要进行全模拟,还要加上一些要素:

```
NOT SIMULATION
VDD 12 0 5V
M1 15 17 20 20 NMOS W=5U L=0.5U AD=12.5P PD=15U AS=20P PS=18U
M2 15 17 12 12 PMOS W=10U L=0.5U AD=25P PD=25U AS=40P PS=36U
RGND 20 0 1U
CLOAD 15 0 100F
.MODEL NFET NMOS<参数表 ...>
.MODEL PFET PMOS<参数表 ...>
...
```

上面第 1 行是电路的名称,同时已经选择 CLOAD 为 100 fF 的外加负载电容。RGND 是一个 1 μΩ 的电阻器把节点 20 下拉至地;另外,可以对网表重新编号或者版图编辑器可以允许提取前在版图上先行定义。⑦

在节点 17 的输入电压可以模拟更多的实际波形。SPICE 的一个有用的结构是图 7.40 所示的脉冲(PULSE)波形。它是由如下语句说明的:

VIN 17 0 PULSE(V1 V2 TD TR TF PW PER)

其中 V1 和 V2 是起始和最终电压,TD 是传输开始之前的时间延迟,TR 是上升时间,TF 是下降时间,PER 是波形重复出现的周期。这使我们能够计算出由低到高和由高到低的过渡时间,这比用阶跃输入求得的要更精确些。另一个有用的波形是指数变化的电源 EXP,它由下列形式来说明:

VIN_EXP 17 0 EXP (V1 V2 TD1 TAU1 TD2 TAU2)

其中 TD1 和 TAU1 是 V1 到 V2 转换的时间延迟和时间常数,而 TD2 和 TAU2 则用于 V2 到 V1 的相反情况。两种情况的时间值都需要仔细选择,这样一个模拟能够通过显示作为时间光滑函数的输出变化,提供有关过渡响应的信息。这些时间值可以通过 RC 模型来估计。

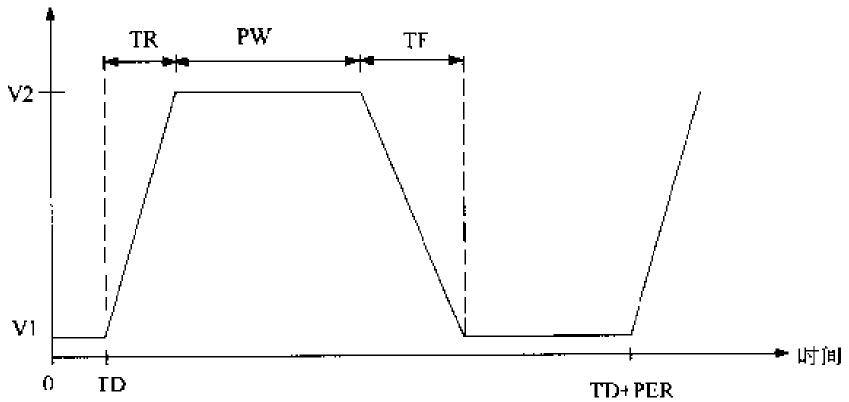


图 7.40 SPICE PULSE 波形

电压传输曲线是由下列带点的命令启动 DC 扫描获得的:

.DC VIN 0 VDD VSTEP

它从 VIN = 0 开始,并按 VSTEP 一步步增加到最终值 VDD。过渡响应由

.TRAN TSTEP TSTOP

计算。这从时间 0 开始且按 TSTEP 的时间单位逐步增长,直到达到时间 TSTOP。这两个命令提供了本章所讨论的电路最关键的工作特性。

同样的技术可以用来模拟任何 CMOS 电路。有一个细节常常会引起混淆,这就是邻近栅之间共用一个公共的有源(n+或 p+)区。它表示为漏还是源是随意的,而总面积和周长可以按照需要在两个 FET 之间分割。必须十分小心,以保证说明两个晶体管的总面积和总周长不

---

⑦ 回想一下,在 SPICE 中接地节点必须编号为节点 0。

超过实际的版图。

【例 7.5】 考虑图 7.41 中的两个 FET。其共享区域的总面积为(10)(8)=80，总周长为2(10+8)= 36。M1 以此作为源区而 M2 以此为漏。其分割方式可写为

M1 ...AS=40P PS=18U
M2 ...AD=40P PD=18U

这是一个等量分配。另一种选择是

M1 ...AS=10P PS=4.5U
M2 ...AD=70P PD=31.5U

其工作情况相同。

在参考资料中可以找到 CMOS 电路 SPICE 模拟技术更多的提示和诀窍。和学习其他编程一样，经验是最好的老师。

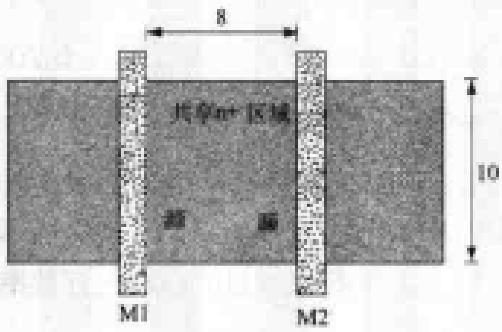


图 7.41 共享有源区

7.10 参考资料

[1] R. Jacob Baker, Harry W. Li, and David E. Boyce, **CMOS Circuit Design, Layout, and Simulation**, IEEE Press, Piscataway, NJ, 1988.

[2] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[3] Yuhua Cheng and Chenming Hu, **MOSFET Modeling & BSIM3 User's Guide**, Kluwer Academic Publishers, Norwell, MA, 1999.

[4] Tor A. Fjeldly, Trond Ytterdal, and Michael Shur, **Introduction to Device Modeling and Circuit Simulation**, John Wiley & Sons, New York, 1998.

[5] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2000.

[6] Jan Rabaey, **Digital Integrated Circuits**, Prentice Hall, Upper Saddle River, NJ, 1996.

[7] Michael Reed and Ron Rohrer, **Applied Introductory Circuit Analy-**

sis, Prentice Hall, Upper Saddle River, NJ, 1999.

[8] Kaushik Roy and Sharat C. Prasad, **Low-Power CMOS VLSI Circuit Design**, Wiley-Interscience, New York, 2000.

[9] Michael John Sebastian Smith, **Application-Specific Integrated Circuits**, Addison-Wesley, Reading, MA, 1997.

[10] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[11] Andrei Vladimirescu, **The SPICE Book**, John Wiley & Sons, New York, 1994.

[12] Gary K. Yeap, **Practical Low Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1998.

7.11 习题

[**7.1**] 一个反相器由以下工艺制造:

$$k'_n = 100\ \mu A/V^2 \qquad V_{Tn} = +0.70\ V$$
$$k'_p = 42\ \mu A/V^2 \qquad V_{Tp} = -0.80\ V \qquad (7.201)$$

电源为 $V_{DD}=3.3$ V。求$(W/L)_n=10$,$(W/L)_p=14$ 时的中点电压 V_M。

[**7.2**] 求电源电压为 3 V,中点电压为 $V_M=1.3$ V 时一个反相器所需要的 β_n/β_p 比值,假设 $V_{Tn}=0.6$ V,$V_{Tp}=-0.82$ V。如果 $k'_n=110\ \mu A/V^2$,迁移率值的关系符合 $\mu_n=2.2\mu_p$,求器件的相对尺寸。

[**7.3**] 一个反相器 FET 的 $\beta_n=2.1\ mA/V^2$,$\beta_p=1.8\ mA/V^2$。其阈值电压为 $V_{Tn}=0.60$ V,$V_{Tp}=-0.70$ V,电源电压为 $V_{DD}=5$ V。输出节点上 FET 的寄生电容估计为 $C_{FET}=74$ fF。

(a) 求中点电压 V_M。

(b) 求 R_n 和 R_p。

(c) 计算当 $C_L=0$ 时输出端的上升和下降时间。

(d) 计算当一个 $C_L=115$ fF 的外加负载连至输出端时的上升和下降时间。

(e) 画出 t_r 和 t_f 与 C_L 的函数关系曲线。

[**7.4**] 根据图 7.11 的反相器版图求中点电压。

[**7.5**] 考虑图 7.11 中的非门,有一个 $C_L=80$ fF 的外加负载连至其输出端。注意其电气沟道长度为 $L=0.8\ \mu m$

(a) 求电路的输入电容。

(b) 求 R_n 和 R_p 的值。

(c) 计算该反相器的上升和下降时间。

[**7.6**] 用 SPICE 模拟图 7.11 电路。假设外加负载为 $C_L=100$ fF,进行 DC 和过渡过程的模拟。

[**7.7**] 有一用相同 nFET 设计的 CMOS NAND2,其中 $\beta_n=2\beta_p$;pFET 的尺寸相同。电源电压 $V_{DD}=5$ V,器件阈值电压为 $V_{Tn}=0.60$ V,$V_{Tp}=-0.70$ V。

(a) 求输入同时切换情况下的中点电压 V_M。

(b) 如果管子的 β 值都相同，反相器的中点电压应为何值？

[**7.8**]　一个 CMOS NOR2 门用具有 β_n 值的 nFET 设计。两个 pFET 都有 $\beta_p = 2.2\beta_n$。如果 $V_{DD} = 3.3$ V, $V_{Tn} = 0.65$ V, $V_{Tp} = -0.80$ V，求输入同时切换情况下的 V_M 值。

[**7.9**]　有一个用相同 nFET 的 NAND3 门，管子宽长比为 4:1。nFET 的工艺互导为 120 $\mu A/V^2$，阈值电压为 0.55 V。电路的电源电压为 5 V。

求输入同时切换情况下中点电压为 $V_M = 2.4$ V 的这个门所需要的 pFETβ_p 值。假设 $V_{Tp} = -0.90$ V, $r = 2.4$。

[**7.10**]　考虑图 P7.1 中的 nFET 链。这是一个 NAND3 门的一部分。已知输出电容值为 $C_{out} = 130$ fF，内部电容为 $C_1 = 36$ fF, $C_2 = 36$ fF。所有的晶体管相同，其工艺为 $\beta_n = 2.0$ mA/V^2, $V_{DD} = 3.3$ V, $V_{Tn} = 0.70$ V

(a) 利用梯形 RC 网络的 Elmore 公式求 $C_{out} = 130$ fF 时的放电时间常数。

(b) 如果忽略 C_1 和 C_2，求时间常数。如果不考虑内部电容，所引起的百分误差是多少？

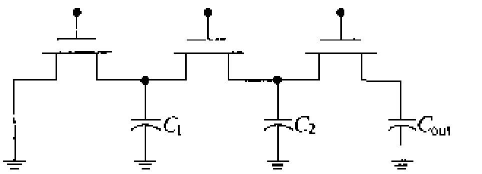


图　P7.1

[**7.11**]　有一复合 CMOS 逻辑门实现以下功能：

$$F = \overline{a \cdot b + c \cdot d \cdot e} \qquad (7.202)$$

(a) 设计该逻辑电路。

(b) 以一个 $\beta_n = \beta_p$ 的反相器作为确定尺寸的参照。如果选择使 pFET 和 nFET 的电阻相等，求该门中的器件尺寸。

[**7.12**]　在一个控制电路中需要一个 CMOS 逻辑门，实现以下功能：

$$F = \overline{x \cdot (y + z) + x \cdot w} \qquad (7.203)$$

(a) 设计该逻辑电路，采用一个 $\beta_n = \beta_p$ 的反相器作为确定尺寸的参照。

(b) 如果选择 pFET 和 nFET 的电阻相同，求门中器件的尺寸。

(c) 假定用与反相器尺寸相同的晶体管，确定将使响应变慢的最坏情况下的 nFET 和 pFET 路径。

[**7.13**]　一个功能为

$$f = \overline{(a + b) \cdot (b + c) \cdot d} \qquad (7.204)$$

的 OAR 采用串－并联 CMOS 结构。

(a) 设计该电路。

(b) 用一个 $\beta_n = 1.5\beta_p$ 的反相器作为确定尺寸的参照，求出能使 nFET 和 pFET 链的路径电阻相等所需要的晶体管尺寸。

(c) 将该功能表达式展开成 AOI 的形式，然后应用同样的确定尺寸的原则设计。哪个设计(AOI 或 OAI)要求的晶体管总面积最小？

[**7.14**]　图 P7.2 中 nFET 的 $\beta_n = 1.50$ mA/V^2，用来作为传输晶体管。工艺使用的 $V_{DD} = 5.0$ V, $V_{Tn} = 0.5$ V。输入端加入逻辑 1 电压 $V_{in} = V_{DD}$，而输出节点的总电容 $C_{out} = 84$ fF。输出电容开始时未充电。

(a) 求逻辑 1 充电时的时间常数。

(b) 计算上升时间，单位用皮秒。

(c) 若输入切换到 $V_{in}=0$ V，计算下降时间。

(d) 用 SPICE 模拟脉冲反应产生输入和输出波形。

[**7.15**] 图 P7.3 中 pFET 传输晶体管的宽长比为 8:1。其工艺为 $k'_p=60\ \mu A/V^2$，$V_{DD}=3.3$ V，$V_{Tn}=-0.8$ V。在时间 $t=0$ 时输出电容充电到电压 V_{DD}，而输入切换到 $V_{in}=0$ V。

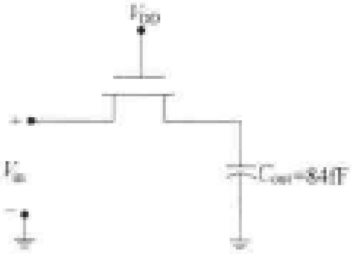


图 P7.2

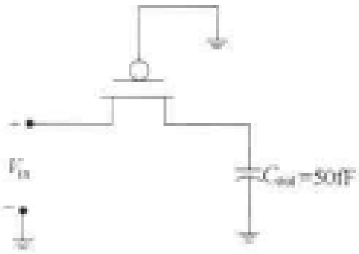
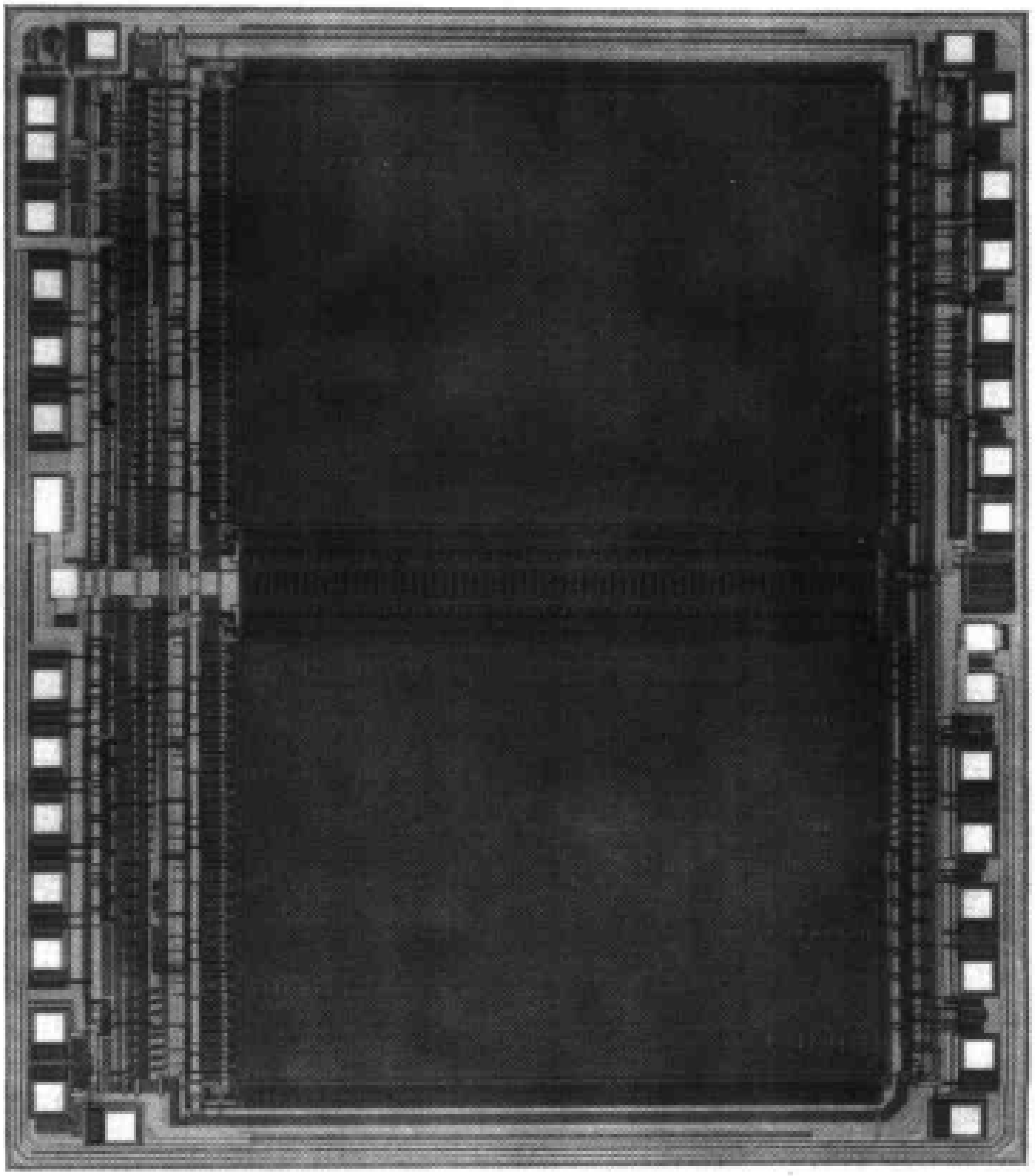


图 P7.3

(a) 求输出节点的下降时间。

(b) 若输入切换回 V_{DD}，求驱动输出电压回复到其高值所需要的上升时间。



第 8 章　高速 CMOS 逻辑电路设计

现代 CMOS 技术已能制造沟道长度小于 0.1 μm 的 MOSFET。FET 的沟道宽度 W 确定了它的宽长比(W/L)，这是决定逻辑电路电气特性的关键参数。

系统设计者必须具有总体观念，即逻辑和体系结构特点排在首位，而电路则是选择实现所需要的功能。然而，在 VLSI 中，是否达到系统的时序目标与逻辑电路的开关速度有着紧密的联系。如果电路不能满足时序要求，则可能不得不调整逻辑。

本章将开始研究高速系统设计并且学习选择晶体管尺寸的技术。这些方法无论对库的建设还是对全定制设计都十分有用。本章所介绍的技术是高速 VLSI 设计整体中的一部分，并且集中针对电子学。由于内容的专业性，有些读者在阅读第 1 遍时可跳过本章和下一章，等到需要时再回过头来学习。

8.1　门延时

在上一章发现，图 8.1 中 CMOS 逻辑门的输出开关时间可以用线性表达式来描述：

$$\begin{aligned} t_r &= t_{r0} + \alpha_p C_L \\ t_f &= t_{f0} + \alpha_n C_L \end{aligned} \tag{8.1}$$

式中 C_L 是外加负载电容。只要知道版图几何图形和工艺参数，该方程组就能分析任意一个门的开关性能。VLSI 设计者则面临着相反的问题：他们的责任是选择逻辑链，并且确定每个晶体管的宽长比。系统被分配在一个有限的芯片面积中，运行必须满足时序上的要求。这就促使我们去开发一种结构化的方法，估计 CMOS 门中的逻辑延时。

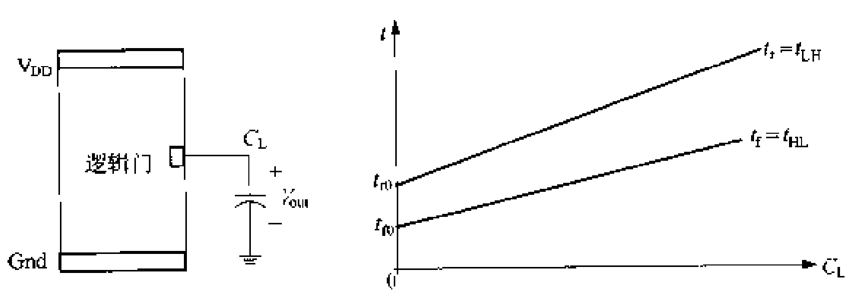


图 8.1　逻辑门输出的开关时间

让我们来考察一种以最小尺寸的 MOSFET 为基础的方法。其版图的几何图形见图 8.2(a)。所画的宽长比(W/L)和有源区的尺寸 X 取决于设计规则。一旦这些条件已知，就能确定器件的寄生参数，且并以它们作为参照。让我们用下标'u'来表示**单位** FET 的参数，于是这

样一个晶体管的电阻为

$$R_u = \frac{1}{k'\left(\frac{W}{L}\right)_u (V_{DD} - V_T)} \tag{8.2}$$

而电容值则为

$$\begin{aligned} C_{Gu} &= C_{ox}(WL)_u \\ C_{Du} &= (C_{GD} + C_{DB})_u \\ C_{Su} &= (C_{GS} + C_{SB})_u \end{aligned} \tag{8.3}$$

在分析中这些都假设为已知参数。为了建立一个设计方法学，将规定所有晶体管的尺寸都是最小宽度 $W_{min} = W_u$ 的整数倍数。图 8.2(b)是 $m = 3$ 的 FET 例子。一般来说，

$$\left(\frac{W}{L}\right)_m = m \left(\frac{W}{L}\right)_u \tag{8.4}$$

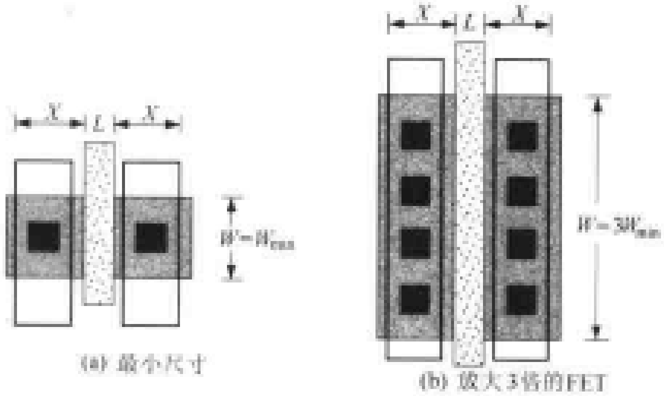


图 8.2 用做参照的单位晶体管

式中，$m = 1,2,3,\cdots$为尺寸说明符。尺寸为 m 的 FET 的电阻和栅电容可以用单位晶体管的参数写成如下：

$$\begin{aligned} R_m &= \frac{R_u}{m} \\ C_{Gm} &= mC_{Gu} \end{aligned} \tag{8.5}$$

我们将放大 FET 的尺寸但使 X 的尺寸仍与单位 FET 相同。对于任意的 m，这意味着漏和源电容近似放大为

$$\begin{aligned} C_{Dm} &= mC_{Du} \\ C_{Sm} &= mC_{Su} \end{aligned} \tag{8.6}$$

这些将在以后的分析中作为等式来用。与电阻公式联合起来得到

$$R_m C_m = R_u C_u = \text{常数} \tag{8.7}$$

这在尺寸缩小理论中非常有用。

假设同时用最小尺寸的 nFET 和 pFET 来设计一个反相器，得到的版图如图 8.3(a)所示；注意该设计中 $\beta_n > \beta_p$。电路的上升时间由 pFET 控制，并可表示为

$$t_{ru} = t_{r0} + \alpha_{pu} C_L \tag{8.8}$$

下降时间

$$t_{fu} = t_{f0} + \alpha_{nu} C_L \tag{8.9}$$

由 nFET 参数控制。由于 $R_p > R_n$，$t_{r0} > t_{f0}$，$\alpha_{pu} > \alpha_{nu}$，所以对于一个给定的负载 C_L，$t_{ru} > t_{fu}$。中点电压为

$$V_M = \frac{V_{DD} - |V_{Tp}| + \sqrt{r} V_{Tn}}{1 + \sqrt{r}} \tag{8.10}$$

式中，$r = (\mu_n / \mu_p)$ 为迁移率之比。由于两个晶体管都是最小尺寸的器件，所以这一互补对的输入电容也是最小值

$$C_{in} = 2C_u = C_{min} \tag{8.11}$$

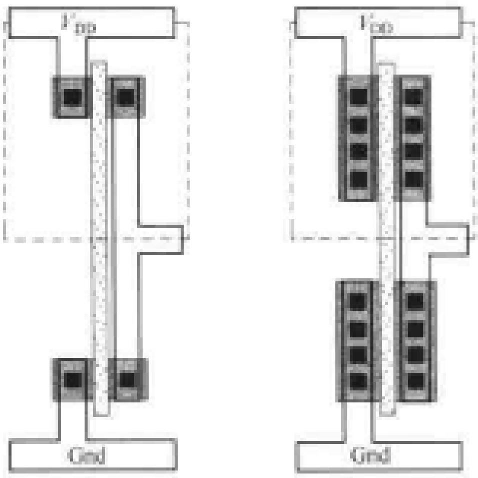
若把 FET 放大 $m = 3$ 倍，得到如图 8.3(b)所示的版图。它的中点电压不变，但开关时间却变了。

为了发现新电路的响应情况，首先注意到零负载的时间 t_{r0} 和 t_{f0} 如方程(8.7)所示(近似)为常数，斜率参数 α 减小为原来的 $(1/m)$，因为电阻减小了同样的倍数。这样

$$t_{r3} = t_{r0} + \frac{\alpha_{pu}}{3} C_L$$
$$t_{f3} = t_{f0} + \frac{\alpha_{nu}}{3} C_L \tag{8.12}$$

上式描述尺寸放大后的电路。该门的输入电容为

$$C_{in} = 3C_{min} \tag{8.13}$$



(a) 单位反相器　(b) $m=3$

图 8.3 用放大的晶体管设计反相器

再考虑图 8.4(a)中的 NAND2 门，它采用最小尺寸晶体管。对于这个电路开关公式必须做一些修改。首先，回想一下零负载时间 t_{r0} 和 t_{f0} 与 C_{FET} 和电阻的乘积成正比。在反相器中，两个 FET 都有电容。由于现在有三个 FET 与输出节点连接，因此引入系数(3/2)乘以内部电容。[①] 电阻则以不同的方式变化。pFET 的电阻 R_p 与反相器的相同，而在输出节点与地之间的 nFET 的电阻 R_n 由于串联的缘故加大了一倍；这使 t_{f0} 和 α_{nu} 都增大两倍。把这些系数包括在公式中得到：

$$t_r = \left(\frac{3}{2}\right) t_{r0} + \alpha_{pu} C_L \quad \text{(单位 NAND2)}$$
$$t_f = 3t_{f0} + 2\alpha_{nu} C_L \tag{8.14}$$

这里忽略了串联 nFET 之间的电容，但确实说明了变化的趋势。由于一对 nFET/pFET 是由最小尺寸器件构成的，输入电容为

$$C_{in} = C_{min} \tag{8.15}$$

① 这里假设 nFET 和 pFET 的电容相等，但实际并非如此，即使它们的尺寸一样。

若把晶体管放大 $m=3$ 倍，如图 8.4(b)所示，则必须对这些公式进行调整。由于电阻减小，两个 a 系数都缩小为$(1/m)$。电阻的减小抵消了 C_{FET}的增加，所以零负载项不变。于是

$$t_r = \left(\frac{3}{2}\right)t_{r0} + \frac{\alpha_{pu}}{3}C_L \tag{8.16}$$

及

$$t_f = 3t_{f0} + \frac{2\alpha_{nu}}{3}C_L \tag{8.17}$$

它们为尺寸放大后的响应时间。输入电容为

$$C_{in} = 3C_{min} \tag{8.18}$$

如果 N 是扇入(输入个数)，那么可以推断，对于用 m 倍尺寸 FET 构成的 N 个输入的 NAND 门，可以写为

$$\left.\begin{aligned} t_r &= \left(\frac{N+1}{2}\right)t_{r0} + \frac{\alpha_{pu}}{m}C_L \\ t_f &= (N+1)t_{f0} + \frac{N\alpha_{nu}}{m}C_L \end{aligned}\right\} \quad \text{(NAND-}N\text{)} \tag{8.19}$$

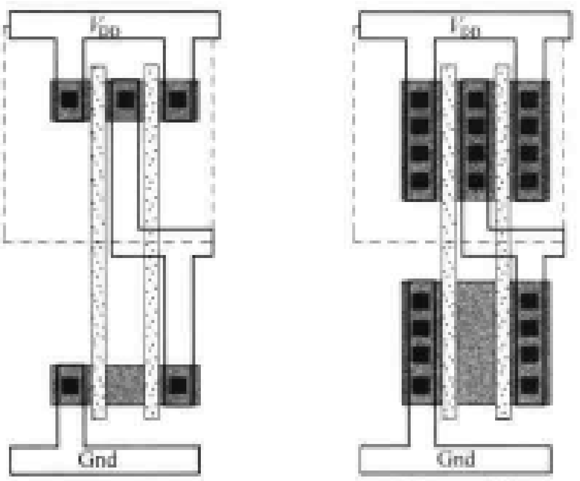
这时输入电容为

$$C_{in} = mC_{min} \tag{8.20}$$

一个二输入的或非门(NOR2)可用同样的方法来分析。图 8.5(a)中单位晶体管版图的开关时间可以近似为

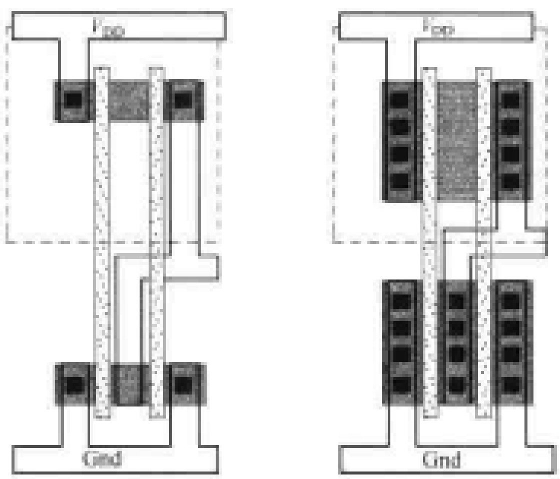
$$\left.\begin{aligned} t_r &= 3t_{r0} + 2\alpha_{pu}C_L \\ t_f &= \left(\frac{3}{2}\right)t_{f0} + \alpha_{nu}C_L \end{aligned}\right\} \quad \text{(单位 NOR2)} \tag{8.21}$$

图 8.5(b)中按 $m=3$ 放大的电路把公式修改为



(a) 单位晶体管　(b) $m=3$ 电路

图 8.4　NAND2 门的放大



(a) 单位晶体管　(b) $m=3$ 电路

图 8.5　NOR 门的放大

$$
\begin{aligned}
t_r &= 3t_{r0} + \frac{2\alpha_{pu}}{3}C_L \\
t_f &= \left(\frac{3}{2}\right)t_{f0} + \frac{\alpha_{nu}}{3}C_L
\end{aligned}
\tag{8.22}
$$

这是由于斜率参数 α 减小的缘故。对于 N 个输入和一般的放大因子 m，上面这些公式可以扩展为 N 个输入 NOR 门的公式：

$$
\begin{aligned}
t_r &= (N+1)t_{r0} + \frac{N\alpha_{pu}}{m}C_L \\
t_f &= \left(\frac{N+1}{2}\right)t_{f0} + \frac{\alpha_{nu}}{m}C_L
\end{aligned}
\quad \text{(NOR-}N\text{)}
\tag{8.23}
$$

同时输入电容为：

$$
C_{in} = mC_{min} \tag{8.24}
$$

这些公式清楚地表明开关时间和输入电容下与以下两方面有关：

- 输入的数目 N(扇入)
- 晶体管尺寸的放大倍数 m

输入电容十分重要，因为它度量一个门对于驱动它的逻辑级的负载有多大。

逻辑门的这个设计技术为估计延时提供了一种结构化的方法。对于一个具有 M 级的逻辑链，可以通过求所有延时的总和，近似通过整个链的总延时：

$$
t_d = \sum_{i=1}^{M} t_i \tag{8.25}
$$

每部分的延时取决于门的类型(即非门，与非门，等等)和它的尺寸，以及在链中下一个门的尺寸和类型。我们仍然需要了解上升和下降时间的差别。

作为一个例子，考虑图 8.6 中的逻辑链，具输入开始为 0 然后过渡到 1。各级的尺寸随 m 值的增加而加大，而输出端为 $C=4\ C_{min}$的电容。总延时为

$$
t_d = t_{NOT}\big|_{m=1} + t_{NAND2}\big|_{m=2} + t_{NOR2}\big|_{m=3} \tag{8.26}
$$

式中，第一和第三项代表下降时间，第二项为上升时间。应用上述公式得到各项为

$$
\begin{aligned}
t_{NOT}\big|_{m=1} &= t_{f0} + \alpha_{nu}2C_{min} \\
t_{NAND2}\big|_{m=2} &= \left(\frac{3}{2}\right)t_{r0} + \frac{\alpha_{pu}}{2}3C_{min} \\
t_{NOR2}\big|_{m=3} &= \left(\frac{3}{2}\right)t_{f0} + \frac{\alpha_{nu}}{3}4C_{min}
\end{aligned}
\tag{8.27}
$$

所以全链的延时为

$$
\begin{aligned}
t_d &= \left(\frac{5}{2}\right)t_{f0} + \left(\frac{10}{3}\right)\alpha_{nu}C_{min} + \left(\frac{3}{2}\right)t_{r0} + \left(\frac{3}{2}\right)\alpha_{pu}C_{min} \\
&= \frac{1}{2}(5t_{f0} + 3t_{r0}) + \left[\left(\frac{10}{3}\right)\alpha_{nu} + \left(\frac{3}{2}\right)\alpha_{pu}\right]C_{min}
\end{aligned}
\tag{8.28}
$$

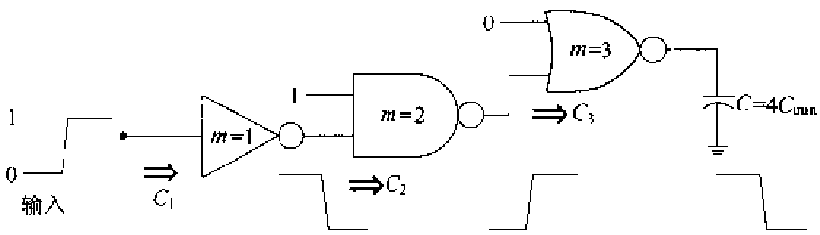


图 8.6 延迟时间举例

必须注意,如果输入不同,t_d 的表达式也将改变。总之,该技术得以用一种统一的方式来估计通过逻辑链的延时。

虽然以上分析是用最小尺寸的 nFET 和 pFET 晶体管进行的,但是很容易把它修改成适用于 $\beta_n=\beta_p$ 的对称设计。这时,对于 $W_n=W_{min}$ 和 $W_p=rW_{min}$ 的反相器电路,反相器的上升和下降时间相等,并由下式得到

$$t_s = t_0+\alpha C_L \tag{8.29}$$

输入电容增加为

$$\begin{aligned} C_{in} &= C_u(1+r) \\ &= C_{inv} \end{aligned} \tag{8.30}$$

现在用它作为参照。把非门中的晶体管放大 m 倍得到

$$t_s = t_0+\frac{\alpha}{m}C_L \tag{8.31}$$

这正如前面一章讨论的那样。多输入门如与非门和或非门电路的分析可用同样的方式进行。注意,若用同样的 m 值来放大 nFET 和 pFET,那么对于 $N>1$ 的门来说,上升和下降时间将不相等。只有在两类 FET 的尺寸不同时才能得到相同的开关时间。如果并联 FET 的尺寸加大 m 倍,则串联晶体管必须加大 mN 倍才能获得对称的设计。

还开发了其他一些方法来估计通过逻辑链的延时。一个简单的办法是以一个最小尺寸的反相器作为基础,然后建立与非门和或非门以增加输入的数目 N。如果画出开关延时作为负载电容 C_L 函数的曲线,就可以得到如图 8.7 所示的变化趋势。根据定义,一个反相器用 $N-1$ 的曲线来描述,并且它是写出如下延时公式的基础:

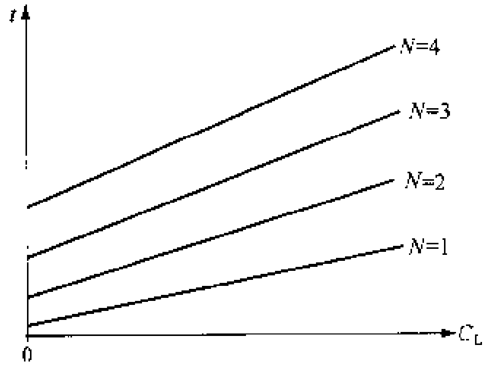


图 8.7 延迟时间与扇入 N 的关系

$$t_d = (A+Bn)\tau_{min} \tag{8.32}$$

其中 A 和 B 是无量纲的常数,

$$\tau_{min} = R_{min}C_{min} \tag{8.33}$$

是最小尺寸反相器的时间常数,而

$$n = \frac{C_L}{C_{min}} \tag{8.34}$$

是由这一级驱动的最小负载因子的数目。这些都来自实验测定的量,即曲线拟合参数。另一种方法是它们可以通过电路模拟来产生。如果扇入增加到 $N=2$(即 NAND2 门或 NOR2 门),则最坏情形下的延时具有较大的零负载值和一个更陡的斜率。当增加到 $N=3$ 时,情况更甚。

一种经验拟合的方法是通过把因子 x_1 乘 t_d 得到的，即考虑延时随 N 用以下形式增加：

$$t_{d,N} = (x_1)^{(N-1)}(A+Bn)\tau_{min} \tag{8.35}$$

例如，从 $N=1$ 到 $N=2$ 每个输入引起延时的增加是 17%，这意味着 $x_1=1.17$，并且

$$t_{d,N} = (1.17)^{(N-1)}(A+Bn)\tau_{min} \tag{8.36}$$

在实际中，将使用多个比较的平均值。如果晶体管按因子 $m=1,2,\cdots$ 来放大，可把这个公式修改为

$$t_{d,N}^{m} = (x_1)^{(N-1)}\left(A+\frac{B}{m}n\right)\tau_{min} \tag{8.37}$$

以考虑驱动强度的增加。同时，对于一个复杂的 N 个输入的逻辑门来说，因为内部电路的电容将增加而减慢了充电和放电过程，所以延时甚至更大。这时，需要乘以另一个经验参数 $x_2>1$得到

$$t_{d,N}^{m} = x_2(x_1)^{(N-1)}\left(A+\frac{B}{m}n\right)\tau_{min} \tag{8.38}$$

在实际中，我们可以认为由于附加 FET 寄生参数的影响，延时将有 5%～20%左右的增加。

虽然这一方法在本质上是近似的，但它确实反映了开关时间随扇入数增加的这一物理事实。若用统一的方法来估计门的延时，那么可对通过各种不同逻辑链的延时进行比较。实际的数值并不精确，但认为它们的相对值是很有价值的。

【例 8.1】 让我们将公式应用于图 8.6 中的逻辑链。三项分别为

$$\begin{aligned} t_{NOT}\big|_{m=1} &= (A+B2)\tau_{min} \\ t_{NAND2}\big|_{m=2} &= x_1\left(A+\frac{B}{2}3\right)\tau_{min} \\ t_{NOR2}\big|_{m=3} &= x_1\left(A+\frac{B}{2}4\right)\tau_{min} \end{aligned} \tag{8.39}$$

这里 NAND2 和 NOR2 都被认为具有相同的最坏情况的延时。全链的延时为

$$\frac{t_d}{\tau_{min}} = [x_1+1]A+\left[\left(\frac{7}{2}\right)x_1+2\right]B \tag{8.40}$$

如果 $x_1=1.17$，则

$$\frac{t_d}{\tau_{min}} = 2.17A+6.1B \tag{8.41}$$

它是与单个反相器延时的比。

正如在后面几章中将要看到的，能够估计通过逻辑链的延时是高速设计的一项重要技巧。在实际数字系统设计中，常能发现截然不同的公式或算法可以产生相同的结果。但是它们使用的是不同类型的逻辑链。这样的技术为决定速度最快的设计提供了基础。

8.2 驱动大电容负载

高速设计的许多要点可以通过对反相器电路延时特性的研究来获得。这些分析为几种熟知的设计技术奠定了基础,它们可以推广用于任意的逻辑门。

考虑图 8.8 中的非门,电路驱动外部负载电容 C_L;晶体管的内部寄生电容 C_{FET}未在图中明确画出。电气特性取决于 β_n 和 β_p 的值。为了简化分析,将采用 $\beta_n=\beta_p=\beta$ 的对称设计。由于 $\beta=k'(W/L)$,这意味着宽长比的关系为

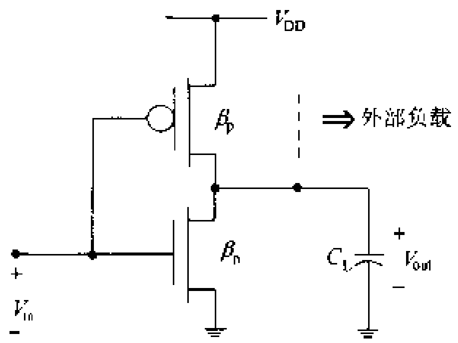


图 8.8 CMOS 反相器电路

$$\left(\frac{W}{L}\right)_p = r\left(\frac{W}{L}\right)_n \tag{8.42}$$

式中 r 是迁移率的比

$$r=\frac{\mu_n}{\mu_p}=\frac{k'_n}{k'_p}>1 \tag{8.43}$$

假设阈值电压的数值相同,即 $V_{Tn}=|V_{Tp}|=V_T$,于是 FET 电阻相等:

$$R_n=R_p=R=\frac{1}{\beta(V_{DD}-V_T)} \tag{8.44}$$

这一设计产生的 VTC 的中点电压为 $V_M=(V_{DD}/2)$并且上升时间等于下降时间。当输出从 0 转换为 1 时,通过 C_L 的电压 $V_{out}(t)$为

$$V_{out}(t)=V_{DD}[1-e^{-t/\tau}] \tag{8.45}$$

而输出从 1 到 0 的切换可描述为

$$V_{out}(t)=V_{DD}e^{-t/\tau} \tag{8.46}$$

两个式子中的时间常数均为以下的积:

$$\tau=RC_{out}=R(C_{FET}+C_L) \tag{8.47}$$

所以开关时间延迟的一般形式为

$$t_s=t_0+\alpha C_L \tag{8.48}$$

式中 t_0 为零负载延迟,α 是 t_s 与 C_L 关系曲线的斜率。对于电路中的变化 t_0 的值几乎不变,而 α 与电阻 R 成正比:

$$\alpha\propto R=\frac{1}{\beta(V_{DD}-V_T)} \tag{8.49}$$

可以选择 β 的值来满足过渡响应的要求。

反相器级的一个重要特性是输入电容 C_{in},它就是 nFET 和 pFET 的栅电容之和:

$$\begin{aligned}C_{in}&=C_{Gn}+C_{Gp}\\&=C_{ox}(A_{Gn}+A_{Gp})\end{aligned} \tag{8.50}$$

其中 A_{Gn}和 A_{Gp}为各个器件的栅面积。假设两个器件的沟道长 L 相等，如果忽略栅的重叠部分 L_0 并近似 $L = L'$ 则

$$\begin{aligned} C_{in} &= C_{ox}L(W_n + W_p) \\ &= (1+r)(C_{ox}LW_n) \\ &= (1+r)C_{Gn} \end{aligned} \tag{8.51}$$

式中的第 2 行利用了公式(8.42)。

假设利用反相器来驱动一个同样的门(反相器)，如图 8.9 所示。这时，由门 1 看到的负载 C_{L1}为

$$C_{L1} = C_{in} \tag{8.52}$$

由于负载电容等于该门自身的输入电容，称它为一个**单位负载值**。开关时间为

$$t_{s1} = t_0 + \alpha C_{in} \tag{8.53}$$

当用这个门来驱动其他负载时，上式作为分析该门性能时的参照是十分方便的。

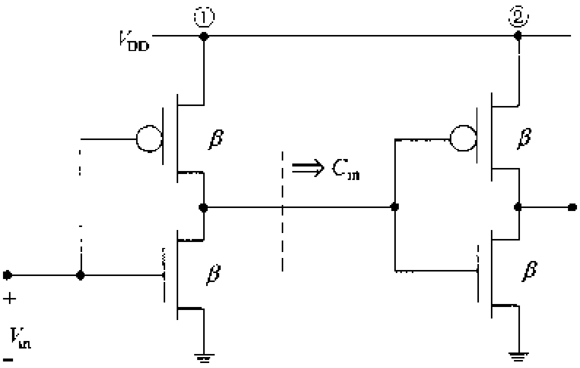


图 8.9　单位负载的概念

如果负载电容增加到一个很大的值 $C_L \gg C_{in}$，那么开关时间也按比例增加。为了保持 t_s 较小，可用较大的晶体管降低电阻来缩小 α 的值。增加 β 值可以补偿较大的负载，这也说明了速度与面积间的这个"互换"关系。假设宽长比按比例因子 $S>1$ 增加，新器件的互导为

$$\beta' = S\beta \tag{8.54}$$

所以电阻降为

$$R' = \frac{R}{S} \tag{8.55}$$

斜率也下降到一个新的值

$$\alpha' = \frac{\alpha}{S} \tag{8.56}$$

这些合在一起得到新反相器的开关时间公式

$$t_s = t_0 + \left(\frac{\alpha}{S}\right)C_L \tag{8.57}$$

补偿因子$(1/S)$使我们可以驱动较大的 C_L 值。如果负载的值为 $C_L = SC_{in}$，则开关时间与单位负载时的开关时间相等。加大晶体管也会影响输入电容，因为

$$W_n' = SW_n \tag{8.58}$$

增加了门的面积。新的电容值为

$$C_{in}' = SC_{in} \tag{8.59}$$

即它也增加同样的放大倍数 S。这就引起了如图 8.10 所示的另一个问题。大的输入电容对驱动电路的作用如一个负载 $C_{L,d} = C_{in}$。为了补偿这一影响，必须加大驱动门中晶体管的尺寸，从而增加它的 $C_{in,d}$电容值。这又使它更难驱动。显然，需要有一种方法学来非常有效地解决这个问题。

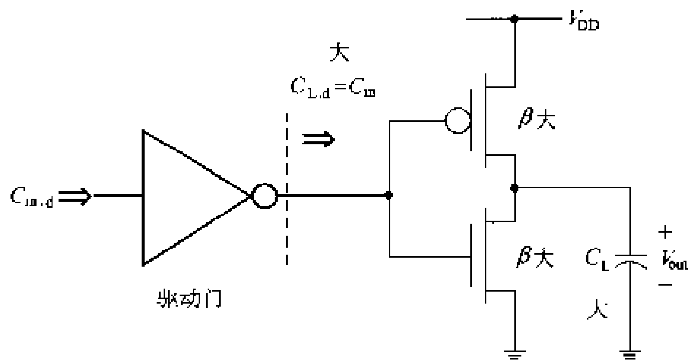


图 8.10 驱动一个具有大输入电容的门

在反相器链中使延时最小

我们将分析的一般问题显示于图 8.11。一个大负载电容 C_L 由一个大反相器门(N)来驱动,这个门又由一个较小的门($N-1$)来驱动,以此类推。第 1 级(1)是一个"标准尺寸"反相器,用它作为参照电路,它具有已知参数:

C_1 = 输入电容

R_1 = FET 电阻

β_1 = 器件互导

各级单调放大,即 1 为最小,N 为最大:

$$\beta_1 < \beta_2 < \beta_3 < \cdots < \beta_{N-1} < \beta_N \tag{8.60}$$

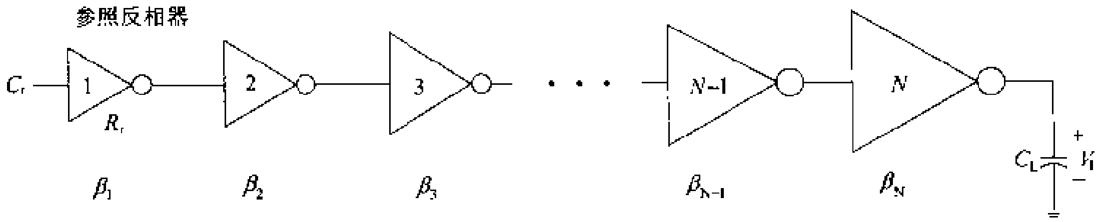


图 8.11 反相器链分析

非门符号的大小也做了调整以显示它们的相对尺寸。最简单的放大方法是按同一因子 $S>1$逐级加大晶体管的尺寸。这意味着

$$\begin{aligned} \beta_2 &= S\beta_1 \\ \beta_3 &= S\beta_2 \end{aligned} \tag{8.61}$$

等等。一般表达式为

$$\beta_{j+1} = S\beta_j \tag{8.62}$$

它表示了第 j 级和第($j+1$)级的关系。

现在的主要问题是:当一个信号加在反相器 1 的输入端时,希望找出级数 N 和放大因子 S,使信号到达负载 C_L 所需要的时间最短。这可通过先研究一个典型级的特性,然后将这些结果应用到整个链来解决。

首先注意在尺寸放大时若用 β_1 作为参照,则意味着

$$\begin{aligned}\beta_2 &= S\beta_1\\ \beta_3 &= S\beta_2 = S^2\beta_1\\ \beta_4 &= S\beta_3 = S^3\beta_1\end{aligned} \tag{8.63}$$

或一般地,对于 $j=2$ 至 N,

$$\beta_j = S^{(j-1)}\beta_1 \tag{8.64}$$

输入电容随 β_j 加大,所以

$$C_j = S^{(j-1)}C_1 \tag{8.65}$$

是第 j 级的输入电容值。FET 电阻减小为 $(1/\beta_j)$,所以得到第 j 级的电阻为

$$R_j = \frac{R_1}{S^{(j-1)}} \tag{8.66}$$

让我们来计算一个典型级的特性。图 8.12 是该链中的第 j 级和第 $j+1$ 级。图中画出了第 j 级的充电电流 i_{ch} 和放电电流 i_{dis}。如果做一个简化假设,使负载电容 $C_{j+12} \gg C_{FET}, j$,则对 $j=1$ 至 N,第 j 级的时间常数为

$$\tau_j = R_jC_{j+1} \tag{8.67}$$

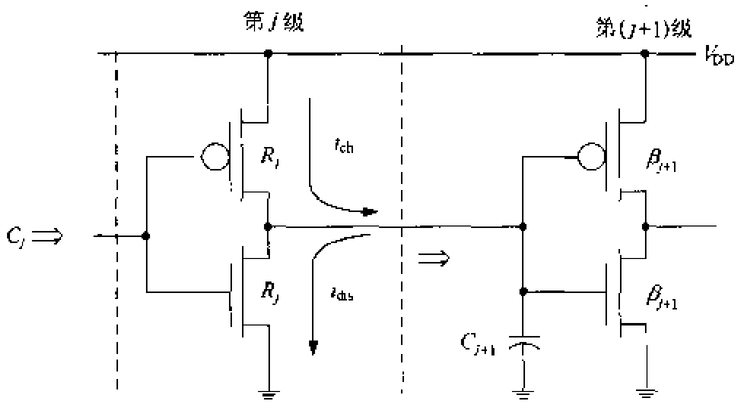


图 8.12 反相器链中一个典型级的特性

这个结果可以用来分析通过该链的总延时。图 8.13 显示了每级的时间常数。该链的总时间常数只要把每项加起来计算即可:

$$\begin{aligned}\tau_d &= \tau_1+\tau_2+\tau_3+\ldots+\tau_{N-1}+\tau_N\\ &= R_1C_2+R_2C_3+R_3C_4+\ldots+R_{N-1}C_N+R_NC_L\end{aligned} \tag{8.68}$$

式中包括了 N 级的负载 C_L,并将它表示成

$$\begin{aligned}C_L &= C_{N+1}\\ &= S^NC_1\end{aligned} \tag{8.69}$$

上式中加入第 2 步是为了与图中的编号方式一致。代入公式(8.65)和(8.66)的尺寸放大关系

得到

$$\tau_d = R_1SC_1 + \frac{R_1}{S}S^2C_1 + \frac{R_1}{S^2}S^3C_1 + \ldots + \frac{R_1}{S^{N-2}}S^{N-1}C_1 + \frac{R_1}{S^{N-1}}S^NC_1 \quad (8.70)$$

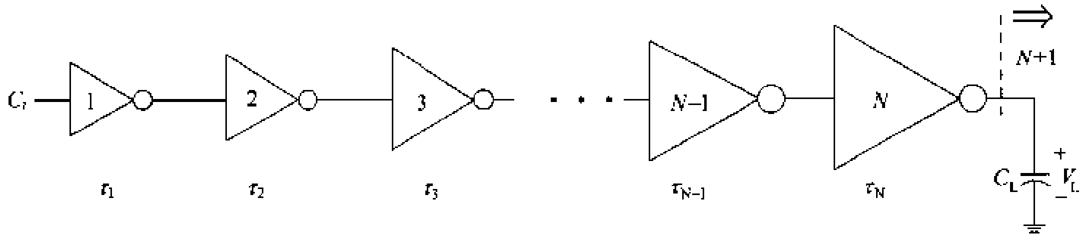


图 8.13 反相器链中的时间常数

由于式中各项都相等,化简得到

$$\begin{aligned}\tau_d &= SR_1C_1 + SR_1C_1 + SR_1C_1 + \ldots + SR_1C_1 + SR_1C_1 \\ &= N(SR_1C_1)\end{aligned} \quad (8.71)$$

所以得到总延时为

$$\tau_d = NS\tau_r \quad (8.72)$$

式中 $\tau_r = R_1C_1$ 是参照的时间常数。这是一个应当记住的非常重要的结果;定性地说,它告诉我们应使各级的信号延时相等。

让我们来考虑延时最小的问题。N 和 S 是未知的,所以需要两个方程。一个是求总延时的方程(8. 72)。另一个可由方程(8. 69)推出,它是本问题的边界条件。首先从下式开始

$$C_L = S^NC_1 \quad (8.73)$$

两边同除以 C_1 并取自然对数得到

$$\ln(S^N) = \ln\left(\frac{C_L}{C_1}\right) = N\ln(S) \quad (8.74)$$

上式也可以写成

$$N = \frac{\ln\left(\frac{C_L}{C_1}\right)}{\ln(S)} \quad (8.75)$$

这是所需要的第二个方程。代入方程(8.72),于是得到以下形式的延迟时间常数

$$\tau_d = \tau_r\ln\left(\frac{C_L}{C_1}\right)\left[\frac{S}{\ln(S)}\right] \quad (8.76)$$

它只是 S 的函数。为了使 τ_d 最小,我们利用求导条件

$$\frac{\partial\tau_d}{\partial S} = \frac{\partial}{\partial S}\left[\frac{S}{\ln(S)}\right] = 0 \quad (8.77)$$

求微分得到

$$\frac{1}{\ln(S)} - \frac{S}{S[\ln(S)]^2} = 0 \tag{8.78}$$

即

$$\ln(S) = 1 \tag{8.79}$$

这是一个非常有意思的式子,因为它的解为

$$S = e \tag{8.80}$$

即 $e = 2.71...$ 是最小延时链的尺寸放大因子。这一设计的级数为

$$N = \frac{\ln\left(\frac{C_L}{C_1}\right)}{\ln(S)} = \ln\left(\frac{C_L}{C_1}\right) \tag{8.81}$$

通过链的总延时为

$$\tau_d = e\ln\left(\frac{C_L}{C_1}\right)\tau_r \tag{8.82}$$

这是反相器链总延时的最小值。解题完毕。

【例 8.2】 为了了解如何应用这些结果,假设需要驱动一个 $C_L = 10$ pF($1\text{pF} = 10^{-12}$F)的负载电容。输入级定义为 $C_1 = 20$ fF $= 20\times10^{-15}$F, $\beta_1 = 200\ \mu\text{A/V}^2$。使延时最小所需要的级数 N 可计算为:

$$N = \ln\left(\frac{10\times10^{-12}}{20\times10^{-15}}\right) = \ln(500) \tag{8.83}$$

由于 $\ln(500) \approx 6.21$,将选择 $N = 6$ 得到一个非反相的链。如果级数确实用求 N 的公式,那么得到的尺寸放大因子 $S = e$。但是,由于把 N 近似为一个(有用的)整数值,可以重新整理公式(8.73),使它成为下列形式以求得更为准确的尺寸放大因子:

$$S = \left(\frac{C_L}{C_1}\right)^{1/N} \tag{8.84}$$

在此例子中

$$S = (500)^{1/6} = 2.82 \tag{8.85}$$

它比理想值稍大一些。

这个设计由 6 个反相器组成,其器件互导分别为

$$\begin{aligned} \beta_2 &= (2.82)\beta_1 \\ \beta_3 &= (2.82^2)\beta_1 = (8)\beta_1 \\ \beta_4 &= (2.82^3)\beta_1 = (22)\beta_1 \\ \beta_5 &= (2.82^4)\beta_1 = (63)\beta_1 \\ \beta_6 &= (2.82^5)\beta_1 = (178)\beta_1 \end{aligned} \tag{8.86}$$

这里我们都取了最接近的整数。注意 FET 的尺寸在接近输出级时增长得很快。

上述理想化的计算可能低估了尺寸放大因子 S,这是因为这一分析忽略了 FET 寄生电容的存在。在实际中,$S>e$,并且它的值与工艺过程有关。为了了解 S 加大的原因,让我们把晶体管寄生电容包括进来重新计算。

图 8.14 显示的是第 j 级的电路,它的输出端包括了 FET 寄生电容 $C_{F,j}$。由于晶体管必须同时驱动 $C_{F,j}$和C_{j+1},这一级的时间常数现在由以下的时间常数给出

$$\tau_j = R_j(C_{F,j} + C_{j+1}) \tag{8.87}$$

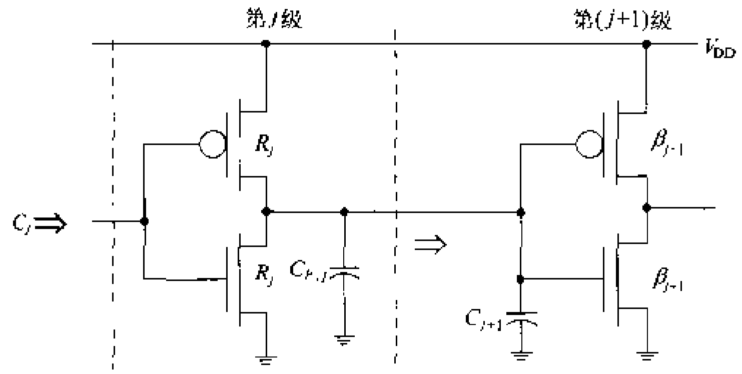


图 8.14 具有内部 FET 电容的驱动器链

FET 寄生电容正比于晶体管的宽长比,所以尺寸放大后的关系为

$$C_{F,j} = S^{(j-1)}C_{F,1} \tag{8.88}$$

式中 $C_{F,1}$是第 1 级 FET 的电容。因此整个链的延迟时间常数为

$$\tau_d = R_1(C_{F,1} + C_2) + R_2(C_{F,2} + C_3) + \ldots + R_N(C_{F,N} + C_L) \tag{8.89}$$

运用尺寸放大的关系可知,每级都有一个寄生参数项 $R_1C_{F,1}$,所以总延时为

$$\tau_d = NR_1C_{F,1} + N(SR_1C_1) \tag{8.90}$$

利用求 N 公式(8.75)得到

$$\tau_d = \left[\frac{\tau_x}{\ln(S)} + \tau_r\left(\frac{S}{\ln(S)}\right)\right]\ln\left(\frac{C_L}{C_1}\right) \tag{8.91}$$

式中

$$\tau_x = R_1C_{F,1} \tag{8.92}$$

对 S 求导并设其为 0 得到延时达最小值的条件:

$$S[\ln(S) - 1] = \frac{\tau_x}{\tau_r} \tag{8.93}$$

这是一个超越方程,其解取决于 τ_x 与 τ_r 之比。注意当 $\tau_x=0$ 时,它简化为较简单的方程,其解为 $S=e$。

【例 8.3】 假设 $\tau_x = 0.2\tau_r$。方程为

$$S[\ln(S)-1] = 0.2 \tag{8.94}$$

其解为

$$S \approx 2.91 > e \tag{8.95}$$

当 $\tau_x = 0.5\tau_r$ 时由该方程得到：

$$S \approx 3.18 \tag{8.96}$$

最后，对于 $\tau_x = \tau_r$，得到

$$S \approx 3.59 \tag{8.97}$$

这说明尺寸放大因子取决于寄生参数。

应当记住这一算法能使从输入至输出的时延最小，但也常使求出的管子尺寸太大而不能实现。当考虑寄生参数并且试图针对非常大的输出电容设计因而使尺寸缩小、因子加大时尤为如此。

8.3 逻辑努力(Logical Effort)

自数字 MOS/ VLSI 电路出现以来，确定逻辑链的每级尺寸一直是一种主要的技术。它用来作为设计快速逻辑链的指南，并且提供许多定性的特点可用于常用的电路。

Sutherland 等重新形成了包含在对逐级变化分析中的概念并且把它用来开发称为"**逻辑努力**"的一般化技术。逻辑努力描述逻辑门的特性以及它们如何在逻辑链中相互作用，并且提供使延时最小的技术。它使这个理论除应用于标准的逻辑门如 NAND 和 NOR 之外，还可扩展到包括复杂的逻辑门电路。在本节中，将考察这一方法的基础，学习如何用它来设计高速逻辑链。有兴趣的读者可以参照参考资料[8]，它对这一有用的技术做了全面出色的文字描述。

8.3.1 基本定义

我们的起始点是将一个反相器定义为一个参照门。最简单的方法是采用一个对称的 NOR 门，它的 $\beta_n = \beta_p$，器件宽长比间的关系为：

$$\left(\frac{W}{L}\right)_p = r\left(\frac{W}{L}\right)_n \tag{8.98}$$

两个 FET 之间的重要差别是 r 的值大于 1。图 8.15 显示一个一倍(1X)尺寸设计的参照电路。管子旁边列出了宽长比的相对值(1 和 r)。这个电路对于定义参照电路的任何$(W/L)_n$ 值都适用，但 1 倍(1X)的参照电路在逻辑链中具有最小的尺寸。较大的器件通过放大这个电路得到。例如一个 4 倍(4X)NOT 门的 nFET 和 pFET 尺寸分别为 4 和 $4r$。

一个门的**逻辑努力** g 定义为它的电容对参照门电容的比：

$$g = \frac{C_{in}}{C_{ref}} \tag{8.99}$$

注意参数 g 具有与逻辑努力这一技术相同的名字;为了区分它们,把这一技术看成一个专用名词并且采用大写字母:Logical Effort。对于 1 倍反相器

$$C_{in} = C_{ox}(A_{Gn} + A_{Gp}) \tag{8.100}$$

式中 A_{Gn}和 A_{Gp}是每个栅的面积:

$$A_{Gn} = W_n L \qquad 和 \qquad A_{Gp} = W_p L \tag{8.101}$$

其中 L 是共同的沟道长度。因为 $W_n = rW_p$,

$$\begin{aligned} C_{in} &= C_{ox} L W_n (1 + r) \\ &= C_{Gn}(1 + r) \\ &= C_{ref} \end{aligned} \tag{8.102}$$

它定义了参照输入电容 C_{ref}。因此根据定义,1 倍反相器的逻辑努力为:

$$g_{NOT} = \frac{C_{ref}}{C_{ref}} = 1 \tag{8.103}$$

$g_{NOT} = 1$ 的值提供了比较其他门性能的基础。注意 nFET 栅电容 C_{Gn}是输入电容的基本单位。

电气努力 h 是由如下的电容比定义的:

$$h = \frac{C_{out}}{C_{in}} \tag{8.104}$$

式中 C_{out}是在输出端看到的外部负载电容。在表达式中需要注意的是:在谈及逻辑努力这一技术时,C_{out}与用在本书其余部分的 C_L 相同。在本节中改变这一表示是为了使那些希望进一步深入研究这一技术的读者有一个较为平滑的过渡。电气努力是驱动 C_{out}所需要的电气驱动强度相对于驱动它自身的输入电容 C_{in}所需要的驱动强度的比。

运用图 8.16 所画的电路,可把通过这个反相器的绝对**延迟时间** d_{abs}写成如下形式:

$$d_{abs} = kR_{ref}(C_{p,ref} + C_{out}) \tag{8.105}$$

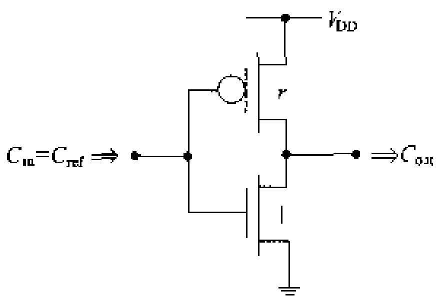


图 8.15 用于计算逻辑努力的参照反相器

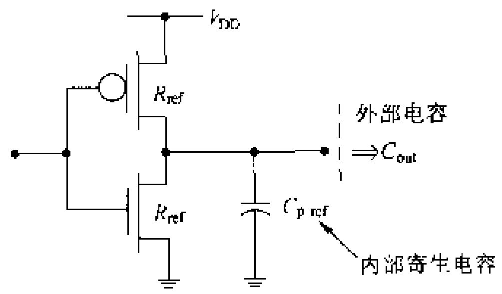


图 8.16 求 1 倍反相器延时的电路

参照 FET 的电阻 R_{ref}对两个管子都相同,因为该设计是对称的。在输出节点上的总电容包括外部 C_{out}值及内部寄生电容 $C_{p,ref}$(也即在表达式中 FET 的电容 C_{FET})。因子 κ 为比例因子;为与第 6 章中的分析相联系,我们可以选择 $\kappa = \ln(9) \approx 2.2$。

现在考虑一个反相器被放大一个倍数 $S>1$。nFET 和 pFET 相对的管子尺寸分别增大为 S 和 rS。FET 电阻减小为

$$R = \frac{R_{ref}}{S} \tag{8.106}$$

而寄生电容增加为

$$C_p = SC_{p,ref} \tag{8.107}$$

于是放大后的门的延时为：

$$\begin{aligned} d_{abs} &= kR(C_p + C_{out}) \\ &= k\frac{R_{ref}}{S}(SC_{p,ref} + C_{out}) \end{aligned} \tag{8.108}$$

现在注意对于放大后的门，输入电容为：

$$C_{in} = SC_{ref} \tag{8.109}$$

把这些项代入，于是就得到

$$\begin{aligned} d_{abs} &= k\frac{R_{ref}}{S}SC_{p,ref} + k\frac{R_{ref}}{S}C_{out} \\ &= kR_{ref}C_{p,ref} + k\frac{R_{ref}}{S}\left(\frac{C_{out}}{C_{ref}}\right)C_{ref} \\ &= kR_{ref}C_{p,ref} + kR_{ref}C_{ref}\left(\frac{C_{out}}{C_{in}}\right) \end{aligned} \tag{8.110}$$

定义参照的时间常数

$$\tau = kR_{ref}C_{ref} \tag{8.111}$$

这可以把延时写成如下形式：

$$d_{abs} = \tau(h + p) \tag{8.112}$$

式中 h 是电气努力，而

$$p = \frac{\tau_{par}}{\tau} = \frac{R_{ref}C_{p,ref}}{R_{ref}C_{ref}} \tag{8.113}$$

是与寄生电容相关的延时。**归一化的延时**

$$d = \frac{d_{abs}}{\tau} = h + p \tag{8.114}$$

是无量纲的，它提供有关这个门的重要信息。在逻辑努力技术中，重点放在求出不同路径的 d。

逻辑努力这一技术所基于的基本概念可从图 8.17 简单的两级反相器电路来理解。总的**路径延时** D 就是各个延时的和，它可以表示为：

$$\begin{aligned} D &= d_1 + d_2 \\ &= (h_1 + p_1) + (h_2 + p_2) \end{aligned} \tag{8.115}$$

式中

$$h_1 = \frac{C_2}{C_1}, \qquad h_2 = \frac{C_3}{C_2} \tag{8.116}$$

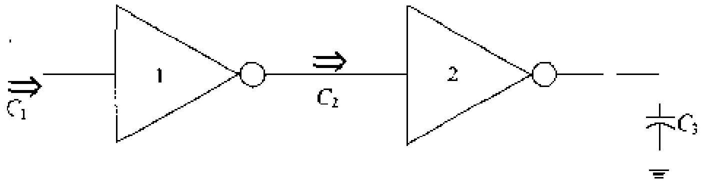


图 8.17 2 级反相器链

为各自的电气努力值。**路径的电气努力** H 定义为如下的比值

$$H = \frac{C_{\text{last}}}{C_{\text{first}}} \tag{8.117}$$

并且可以表示成如下的积：

$$H = h_1 h_2 \tag{8.118}$$

这可以从下式看出：

$$H = \left(\frac{C_2}{C_1}\right)\left(\frac{C_3}{C_2}\right) = \frac{C_3}{C_1} \tag{8.119}$$

这个积的形式是 H 的一般特性。运用

$$h_2 = \frac{H}{h_1} \tag{8.120}$$

则路径延时方程变为

$$D = (h_1 + p_1) + \left(\frac{H}{h_1} + p_2\right) \tag{8.121}$$

逻辑努力技术的主要目的是使通过逻辑链的延时减到最小。就现在的情形，这个条件可以通过计算导数来求得：

$$\frac{\partial D}{\partial h_1} = \frac{\partial}{\partial h_1}\left[(h_1 + p_1) + \left(\frac{H}{h_1} + p_2\right)\right] \tag{8.122}$$

微分式中寄生参数 p_1 和 p_2 是常数，所以

$$\frac{\partial D}{\partial h_1} = 1 - \frac{H}{h_1^2} = 0 \tag{8.123}$$

由于运用 $H = h_1 h_2$，所以该方程表明如果

$$h_1 = h_2 \tag{8.124}$$

则路径延时达到最小值。

因为通过一个反相器的延时正比于 h ，这相当于说，使得通过路径中每级的延时相同可以使这条路径的总延时最小。自然这与较严格分析时所得到的结论相同。

8.3.2 一般化情形

逻辑努力技术的真正功能在于它能被一般化以包括任何 CMOS 逻辑门。这一结论使我们可以估计逻辑链的延时，并且提供实现最小延时设计所需要的尺寸比例关系。

使这一技术一般化的第一步是求出基本 CMOS 门的逻辑努力参数 g。所有的计算都参照一倍的参照反相器，其输入电容为 C_{ref}，管子电阻为 R_{ref}。最简单的设计是保持对称的设计，也就是 $R_n = R_p = R_{ref}$。这要求我们调整串联管子的尺寸。

图 8.18(a)显示一个对称的一倍 NAND2 门。pFET 的尺寸仍为 r，因为最坏情形下从输出至电源的路径与反相器是相同的。然而 nFET 必需是反相器值的 2 倍，因为它们是串联在一起的；它们的相对值因此表示为 2。对每个输入，输入电容于是为

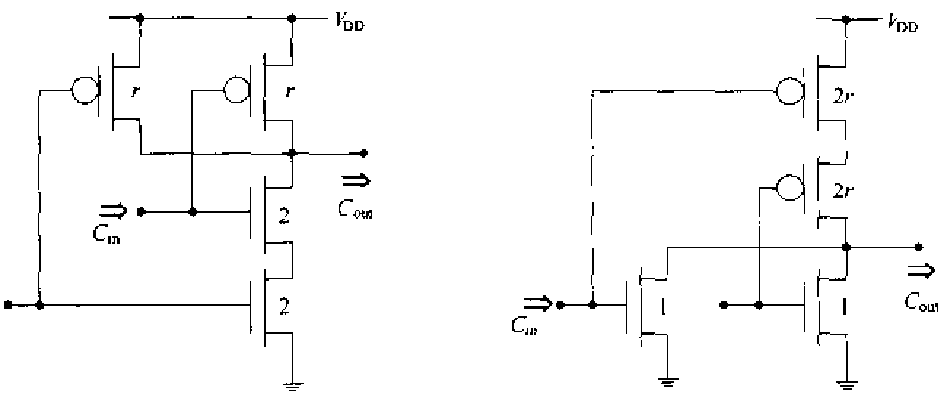
$$C_{in} = C_{Gn}(2+r) \tag{8.125}$$

所以 NAND2 门的逻辑努力是

$$g_{NAND2} = \frac{C_{Gn}(2+r)}{C_{ref}} = \frac{2+r}{1+r} \tag{8.126}$$

这足以描述该门的特性用于延时计算。

图 8.18 (b)的一倍 NOR2 电路可用相同方式分析。并联 nFET 的相对尺寸为 1 而选择 pFET 的尺寸为 $2r$，使 R_p 与 $R_r$cf 相同。因此输入电容为



(a) NAND2　　(b) NOR2

图 8.18　对称 NAND 和 NOR 门

$$C_{in} = C_{Gn}(1+2r) \tag{8.127}$$

所以这个门的逻辑努力是

$$g_{NOR2} = \frac{C_{Gn}(1+2r)}{C_{ref}} = \frac{1+2r}{1+r} \tag{8.128}$$

注意 g 的值与比率 r 有关。

这些结果可以一般化到扇入更多的门。一个 n 个输入的 NAND 门具有尺寸为 r 的 n 个 pFET 并联及尺寸为 n 的 n 个 nFET 串联。在输入端看到的电容为

$$C_{in} = C_{Gn}(n+r) \tag{8.129}$$

所以逻辑努力为

$$g_{\text{NAND}} = \frac{n+r}{1+r} \tag{8.130}$$

一个 n 个输入 NOR 门的逻辑努力为

$$g_{\text{NOR}} = \frac{1+nr}{1+r} \tag{8.131}$$

它可以用相同的方法验证。很容易看到任何基本的 CMOS 门都可以用逻辑努力 g 来描述它的特性。

通过一个一般门的延时可表示为

$$d = gh + p \tag{8.132}$$

逻辑努力参数 g 的主要影响是修改上式第 1 项以考虑在不同门之间驱动特性上的差别。对于具有 N 级的长逻辑链，每个门的特性由以下的延时来表示：

$$d_i = g_i h_i + p_i \tag{8.133}$$

式中 $i=1$ 至 N。**总路径延时** D 是以下的和：

$$D = \sum_{i=1}^{N} d_i = \sum_{i=1}^{N} (g_i h_i + p_i) \tag{8.134}$$

路径逻辑努力 G 就是各个因子的积：

$$G = \prod_{i=1}^{N} g_i = g_1 g_2 \ldots g_N \tag{8.135}$$

而**路径电气努力** H 可以用类似的方式定义如下：

$$H = \prod_{i=1}^{N} h_i = h_1 h_2 \ldots h_N \tag{8.136}$$

这两项合在一起得到**路径努力** F

$$\begin{aligned} F &= GH \\ &= (g_1 h_1)(g_2 h_2)(g_3 h_3)\ldots(g_N h_N) \\ &= f_1 f_2 \ldots f_N \end{aligned} \tag{8.137}$$

如果对于每一个 i，

$$g_i h_i = \text{常数} = \hat{f} \tag{8.138}$$

那么通过逻辑链的延时最小。这一点与我们对于简单的两级反相器链的结论是一致的。因此最优的路径努力为

$$F = \hat{f}^{N} \tag{8.139}$$

最快的设计是每级都有

$$gh = \hat{f} = F^{1/N} \tag{8.140}$$

这是逻辑努力技术的主要方程。根据一个 N 级逻辑链的组成，可以求得 F 值。由此可以确定每级的尺寸使符合最优的电气努力值：

$$h_i = \frac{\hat{f}}{g_i} \tag{8.141}$$

于是最优的路径延时为

$$\hat{D} = NF^{1/N} + P \tag{8.142}$$

式中

$$P = \sum_{i=1}^{N} p_i \tag{8.143}$$

是寄生延时的和。一般地，一个反相器的 p_{ref} 最小，而多输入的门显示了较大的寄生延迟时间。一个简单的估计可以写成：

$$p = np_{ref} \tag{8.144}$$

它是一个 n 输入门的寄生延时。

【例 8.4】 让我们运用逻辑努力技术来分析图 8.19 的逻辑链。假设 $C_4 = 500$ fF，$C_1 = 20$ fF。首先，路径逻辑努力为：

$$G = g_{NOT}\, g_{NOR2}\, g_{NAND2} = (1)\left(\frac{1+2r}{1+r}\right)\left(\frac{2+r}{1+r}\right) \tag{8.145}$$

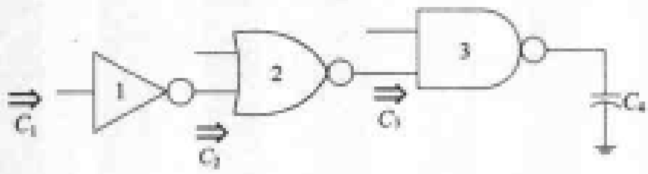


图 8.19　例 8.4 的逻辑链

假设 $r = 2.5$，计算可得

$$G = (1)\left(\frac{6}{3.5}\right)\left(\frac{4.5}{3.5}\right) = 2.2 \tag{8.146}$$

路径电气努力为

$$H = \frac{C_4}{C_1} = \frac{500}{20} = 25 \tag{8.147}$$

所以路径努力为

$$F = GH = 55 \tag{8.148}$$

最优的每级努力为

$$\hat{f} = F^{1/N} = (55)^{1/3} = 3.8 \tag{8.149}$$

由上式得出总的路径延时为

$$\begin{aligned}\hat{D} &= 3(3.8) + P \\ &= 11.41 + P\end{aligned} \tag{8.150}$$

式中

$$P = (p_{NOT} + p_{NOR2} + p_{NAND2}) \tag{8.151}$$

是由具体工艺决定的寄生延迟项。

确定尺寸的方程可以通过分析被优化的量得到。从输出端的 NAND2 门开始，由 $g_{NAND2} = (4.5/3.5) = 1.29$，得到

$$h_3 = \frac{3.8}{1.29} = 2.95 = \frac{C_4}{C_3} \tag{8.152}$$

所以

$$C_3 = \frac{500}{2.95} = 169.5\ \text{fF} \tag{8.153}$$

因为 C_3 是 NAND2 门的输入电容，可以运用方程(8.125)写出尺寸放大后的门满足：

$$\begin{aligned}C_3 &= S_3 C_{Gn}(2 + r) \\ &= S_3(4.5 C_{Gn})\end{aligned} \tag{8.154}$$

式中 S_3 是尺寸放大因子。

NOR2 门用同样的方式分析。因为 $g_{NOR2} = 1.17$，有

$$h_2 = \frac{3.8}{1.71} = 2.22 = \frac{C_3}{C_2} \tag{8.155}$$

于是

$$C_2 = \frac{169.5}{2.22} = 76.35\ \text{fF} \tag{8.156}$$

NOR2 门的输入电容为

$$\begin{aligned}C_2 &= S_2 C_{Gn}(1 + 2r) \\ &= S_2(6 C_{Gn})\end{aligned} \tag{8.157}$$

输入端的 NOT 门定义逻辑努力为 1，所以

$$h_1 = \frac{3.8}{1} = \frac{C_2}{C_1} \tag{8.158}$$

于是得到 $C_1 = (76.35/3.8) = 20$ fF 与所要求的一样。

我们选择的参照门是输入端的 NOT 门，即 $C_1 = C_{ref} = 3.5C_{Gn}$。为了达到最小延时，NOR 和 NAND 门放大为

$$S_2 = \frac{76.35}{(6)(3.5C_{Gn})} = \frac{3.64}{C_{Gn}}$$
$$S_3 = \frac{169.5}{(4.5)(3.5C_{Gn})} = \frac{10.76}{C_{Gn}} \quad (8.159)$$

这些尺寸放大因子值的参照电容为

$$C_{Gn} = \frac{20}{3.5} = 5.71\ \text{fF} \quad (8.160)$$

式中

$$C_{Gn} = C_{ox}W_nL \quad (8.161)$$

由它可求出参照 nFET 的沟道宽度 W_n。

另一个步骤是选择最小尺寸 1 倍的反相器作为参照门。如果一倍门的 $C_{ref}=8$ fF，那么尺寸放大因子为 $S_1=2.5$(对 NOT 门)，$S_2=1.59$ 及 $S_3=4.71$。通常参照门是根据方便来选择的。

8.3.3 级数的优化

CMOS 逻辑链一个熟知的特性是我们常常可以把反相器插入逻辑链中，减少总的延迟时间。这也许与在逻辑设计课程引论中建立的简单直觉不同，但它却是基于这样的一个事实：即把驱动强度分配在几级之间要比计算逻辑符号的数目更为重要。逻辑努力技术通过运用路径延时 D 显示了这一特点。

首先注意到一个反相器的逻辑努力为 $g_{NOT}=1$。由于

$$G = g_1g_2\cdots g_N \quad (8.162)$$

所以把它乘以另外的 g_{NOT}因子并不改变路径努力的数值

$$F = GH \quad (8.163)$$

使延迟时间最小的条件可以表示为

$$\hat{f} = F^{1/N} = (GH)^{1/N} \quad (8.164)$$

所以总的路径延时为

$$\hat{D} = NF^{1/N} + P \quad (8.165)$$

一般地，$F^{1/N}$随 N 的增加而减少。因此通过插入反相器有可能得到较小的路径延时。然而注意由于添加的反相器使在 P 中寄生延时增加将抵消掉一些性能。

【例 8.5】 为了看到这一关系，假设 $F=200$。对于 $N=3$ 有

$$3(200)^{1/3} = 17.54 \quad (8.166)$$

对于 $N=4$，得到

$$4(200)^{1/4} = 15.04 \tag{8.167}$$

对于 $N=5$，得到

$$5(200)^{1/5} = 14.43 \tag{8.168}$$

如果使 $N=10$，那么这一项却会增加：

$$10(200)^{1/10} = 16.99 \tag{8.169}$$

实际上已越过了最优的级数。

这个问题的分析表明，对于给定的 F 最优的级数可以通过求解以下超越方程[8]得到：

$$F^{1/N}[1-\ln(F^{1/N})]+p_{\text{ref}} = 0 \tag{8.170}$$

通过定义

$$\rho = F^{1/N} \tag{8.171}$$

(8.170)式可以改写成一个显得较简单的形式，即

$$\rho[1-\ln(\rho)]+p_{\text{ref}} = 0 \tag{8.172}$$

稍加回想不难确定上式与从电路考虑推导出来的方程(8.93)具有相同的形式，因而说明了这两种方法是等效的。逻辑努力技术的功能在于它并不只局限于反相器。

对于小的 p_{ref}值，近似解为

$$\rho = 0.71p_{\text{ref}}+2.82 \tag{8.173}$$

上式对于在初始设计阶段估计 N 的最优值是非常有用的。

8.3.4 逻辑面积

所占用的芯片面积是一个很重要的考虑因素，特别是在按比例放大尺寸的设计中。根据逻辑努力技术得到的各量，通过把各个 FET 的栅面积相加就可估计该电路要求的面积。对第 i 个门，利用下式可以计算出它的逻辑面积(LA)：

$$LA_i = W_i \times L \tag{8.174}$$

式中，L 是沟道长度而 W_i 由门的尺寸决定。例如，一个一倍 NOT 门的 $L=1$ 个单位，其逻辑面积为：

$$LA_{\text{NOT}} = 1+r \tag{8.175}$$

它同时考虑了 pFET 和 nFET 的尺寸。如果它被扩大 S 倍($S>1$)，那么逻辑面积增加为：

$$LA_{\text{NOT}} = S(1+r) \tag{8.176}$$

类似地，一个扩大 S 倍的 NOR2 门的逻辑面积为

$$LA_{\text{NOR2}} = S(1+2r) \tag{8.177}$$

而对于 NAND2 门，

$$LA_{NAND2} = S(2+r) \tag{8.178}$$

对于具有 M 个门的电路，总的逻辑面积为

$$LA = \sum_{i=1}^{M} LA_i \tag{8.179}$$

这使我们有一个简单的尺度来比较不同设计对面积的要求。然而注意，由于忽略了漏源所占的空间、互连线、阱等等，所以它只是一个粗略的估计。

8.3.5 分支情况

至今逻辑努力技术应用于简单定义的路径。当一个逻辑门驱动两个或更多的门时，数据路径出现分支，必须考虑不在主要路径上但又对主要路径产生负载电容效应的那些门。这个情形表示在图 8.20 的逻辑图中，图中我们所关心的从输入至输出的主要路径用粗黑线表示。跟踪这个电路可以看到两个分支点。在这两个分支情况中，都是 NOR2 门对 NAND2 增加了负载电容，它们不能被忽略。

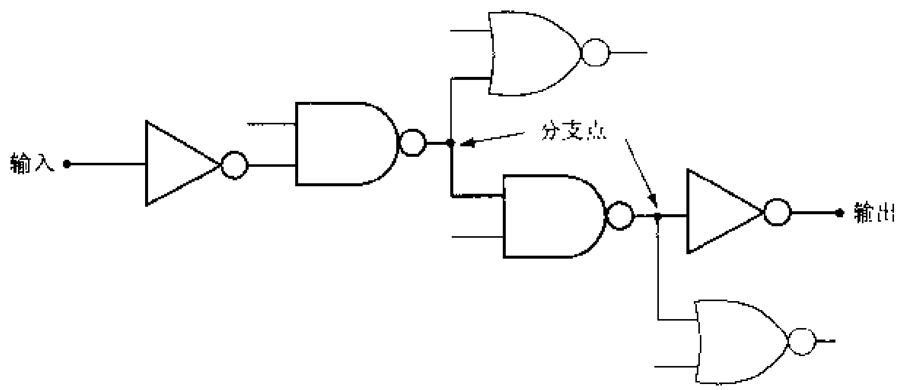


图 8.20 分支情况

这些影响可以通过在每个分支点处引入"**分支努力**"b 加以考虑：

$$b = \frac{C_T}{C_{path}} \tag{8.180}$$

式中，C_{path}是主逻辑路径上的电容，而

$$C_T = C_{path} + C_{off} \tag{8.181}$$

代表了从该节点看到的总电容。在这个方程中，C_{off}包括所有不是主路径上的负载电容。分支努力的特点是 $b>1$，它考虑了附加负载的影响。路径分支努力为下式乘积：

$$B = \prod_{i} b_i \tag{8.182}$$

式中，b_i 是每个分支处的分支努力。

【例 8.6】 考虑图 8.20 的逻辑电路。在第一个分支点处，一个 NAND2 门驱动在主路径上的另一个 NOR2 门以及一个分支路径上的 NOR2 门。假设都是单位门的尺寸，该点的分支努力 b_1 为

$$b_1 = \frac{C_{NAND2} + C_{NOR2}}{C_{NAND2}} = \frac{(2+r)+(1+2r)}{(2+r)} = \frac{3(1+r)}{(2+r)} \tag{8.183}$$

图中第二个分支点的分支努力为

$$b_2 = \frac{C_{NOT} + C_{NOR2}}{C_{NOT}} \tag{8.184}$$

即

$$b_2 = \frac{(1+r)+(1+2r)}{(1+r)} = \frac{(2+3r)}{(1+r)} \tag{8.185}$$

于是所选从输入至输出的路径分支努力为

$$B = \frac{3(1+r)}{(2+r)}\frac{(2+3r)}{(1+r)} = \frac{3(2+3r)}{(2+r)} \tag{8.186}$$

一旦算出路径分支努力，就可以把路径努力 F 的公式修改为：

$$F = GHB \tag{8.187}$$

然后计算过程就可以与没有分支的较简单情形的方式相同。这使我们可以把逻辑努力技术扩展到任意的逻辑形态，并且分析每条路径的相对延迟。

8.3.6 小结

这一逻辑努力的简短讨论说明了这一技术的用途。它在复杂的系统设计中特别有用，因为有几种算法选择而达到同一结果。逻辑努力可以比较各种不同电路的性能，看看哪个对我们的设计更好。这些考虑将在本书的后几章中讨论。

8.4 BiCMOS 驱动器[②]

BiCMOS 是一种改型的 CMOS 工艺，它还包括双极型管作为电路单元。在数字设计中，

② 可以跳过本节而不影响讨论的连续性。

采用 BiCMOS 电路驱动大电容连线比只有 MOSFET 的驱动电路更为有效。BiCMOS 工艺要比标准的 CMOS 工艺成本高,而且双极型管不可避免地具有本身固有的电压降,因此不希望在低电压应用中使用它们。

8.4.1 双极型管的特性

双极型管(BJT)是一个三端器件,它的电气特性来自于 pn 结的特性。有两类 BJT,npn 和 pnp。流过一个 npn 管的电流大部分是由电子形成的,而流过 pnp 器件的电流则是由空穴引起的。由于电子比空穴快,在高速 BiCMOS 电路中多运用 npn 器件。

npn BJT 的电路符号表示在图 8.21 (a)中。这一器件具有三个终端,它们分别称为基极(B),射极(E)带有箭头,以及集电极(C)。npn BJT 的简化"原型"结构显示在图 8.21 (b)中;图中显示了 npn 层结构,该器件由此得名。该图表明 npn 管子可以看成是两个背靠背的 pn 结二极管,一个在基极和发射极之间,另一个在基极和集电极之间。通过 BJT 的电流由两个电压控制,即基极-射极电压 V_{BE}以及基极－集电极电压 V_{BC},它们偏置了两个 pn 结。当"+"极性标在 p 型的基极层上时,这两个电压都定义为正值。一个正电压表明 pn 结上是正向偏置,因此允许电流流过,而负电压则表明反向偏置。

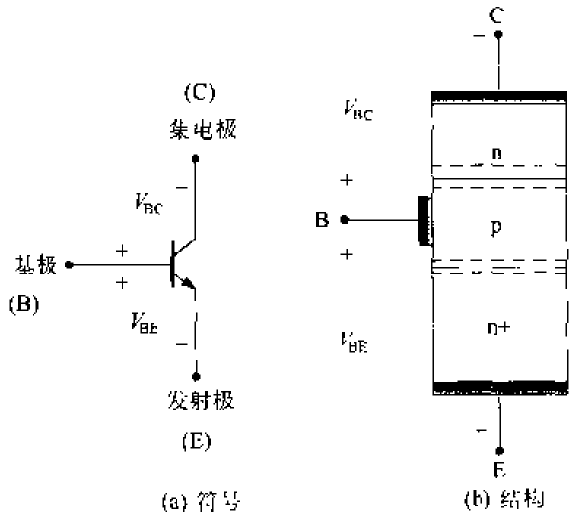


图 8.21 npn BJT 的符号与结构

双极型管的操作比较复杂,因为偏置电压可以是正的或负的(反极性)。考虑图 8.22(a)所示的情形,电流 I_C,I_B 和 I_E 是由偏置电压决定的,但它们极性的每种组合则为不同的工作模式。这些总结在图 8.22(b)中。该图用四个坐标象限表示 V_{BE}和 V_{BC}的极性。**正向导通偏置**定义为 $V_{BE}>0$ 和 $V_{BC}<0$,即基极－发射极的 pn 结正向偏置而基极-集电极的 pn 结反向偏置。这个工作模式可以用于放大及可控电流流动,因而用于模拟电路。反过来的情形即 $V_{BE}<0$和 $V_{BC}>0$,称为**反向导通偏置**,它只用于几种特殊情形。如果两个 pn 结都是正向偏置,即 $V_{BE}<0$ 和 $V_{BC}>0$,那么这个器件处于**饱和**状态。在这一情形中,可以有大电流流过这个器件,但管子却并不控制电流值。应当记住,在 BJT 中谈到的饱和与 FET 饱和没有任何联系。最后一种情形是两个 pn 结都反向偏置,即 $V_{BE}>0$ 及 $V_{BC}<0$,只有很小的漏电流流过,此时 BJT 处于**截止**状态。它可以被模拟成一个开路的开关。

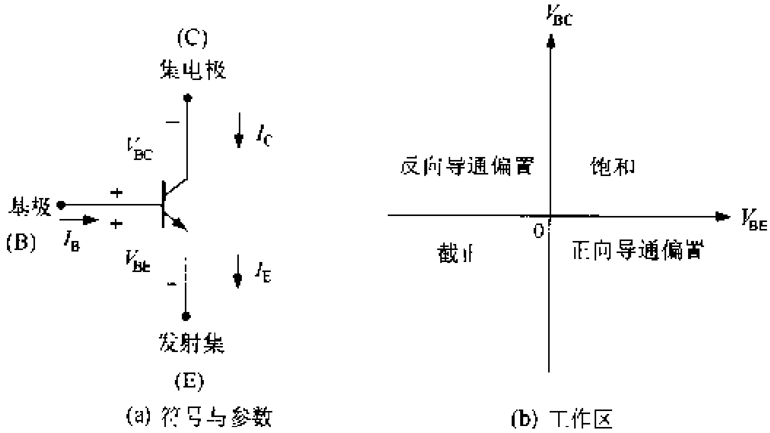


图 8.22 双极型晶体管的工作区

双极型管比 MOSFET 快，但在集成电路中制造比较复杂。让我们来考察一下正向导通偏置的情形，理解为什么双极型管能够提供较快的开关性能。图 8.23(a)显示了采用这一偏置的器件。集电极和射极电流的关系是：

$$I_C = \alpha_F I_E \tag{8.188}$$

式中 $\alpha_F<1$ 是该器件的正向 α；实际中 $\alpha_F-0.99$，所以 I_C 和 I_E 大致相同。图 8.23(b)为在正向导通偏置时的**传输曲线** $I_C(V_{BE})$，它可以用下式描述：

$$I_C \approx I_S e^{V_{BE}/V_{th}} \tag{8.189}$$

式中，I_S 是饱和电流而 V_{th} 是热电势。I_S 的值是由结构和工艺决定的，而热电势在 $T=300$ K 时约为 26 mV 且随温度线性增长。该图显示了当基极－发射极电压达到 $V_{BE(on)}$ 值时，流过的电流变得很显著。$V_{BE(on)}$ 通常估计为 0.5～0.7 V。一旦达到这个值，电流就会随 V_{BE} 的增加而呈指数关系地增长。

考虑图 8.24 中的简单电路。当 BJT 处于正向导通偏置时，流出电容的电流为：

$$I_C = -C_{out}\frac{dV_{out}}{dt} = I_S e^{V_{BE}/V_{th}} \tag{8.190}$$

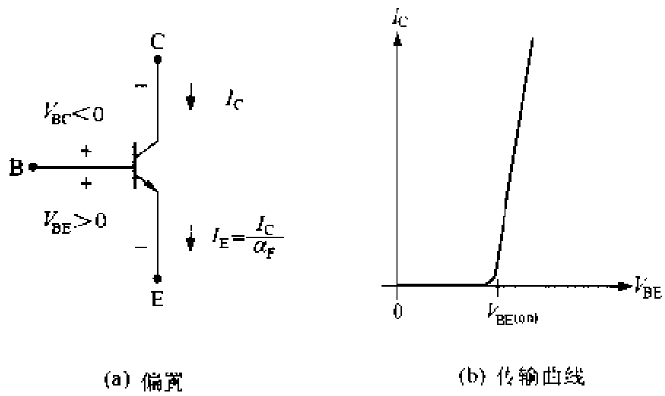


图 8.23 BJT 的正向导通偏置

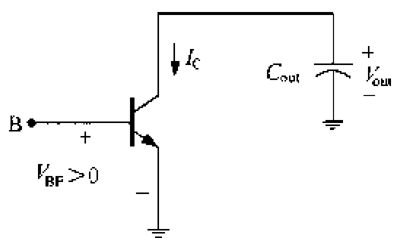


图 8.24 用 BJT 使电容放电

可用下式估计放电时间

$$\Delta t \approx \frac{(\Delta V_{out})}{I_C C_{out}} \tag{8.191}$$

式中，ΔV_{out}是电压的改变。I_C 的值可以很大，很容易达到几十至几百毫安，因而甚至对于较大的 C_{out}值也能使放电时间 Δt 减少。一个双极型管比占同样面积的 FET 可以更快地完成任务，这使 BiCMOS 十分有吸引力。

通过 BJT 的电流是由于粒子扩散的机理，而不是像在 FET 中那样靠电场帮助运动的。图 8.25 概括了这个原型器件的正向导通工作情况。当基极－发射极正向偏置时，电子从发射极流向基极。一旦电子出现在基区，它们就成为少数载流子电荷载体并向集电极扩散。虽然有些电子与空穴碰撞而消失，如果基区的宽度 x_B 足够小(典型情况下小于 0.5 μm)，大多数电子将到达集电极。这就形成了从集电极到发射极的电流。分析表明饱和电流由下式给出：

$$I_S = qA_E \frac{D_n n_i^2}{x_B N_{aB}} \tag{8.192}$$

式中，A_E[cm^2]是发射极面积，D_n[cm^2/s]是电子在基区的扩散系数，它度量了扩散运动的快慢，q 是电子电荷，而 N_{aB}[cm^{-3}]是基区的受子掺杂浓度。饱和电流的典型值为 $I_S = 0.1\ pA = 10^{-13}$A。虽则这一电流非常小，但电流取决于 V_{BE}的指数关系使 I_C的值很大。集成双极管的剖面图显示在图 8.26 中。原型结构可以从发射极 n+ 区下面的中心部分看到。因为需要专门的工艺层来形成这个器件，所以 BiCMOS 芯片的工艺费用要比基本的 CMOS 设计更贵。

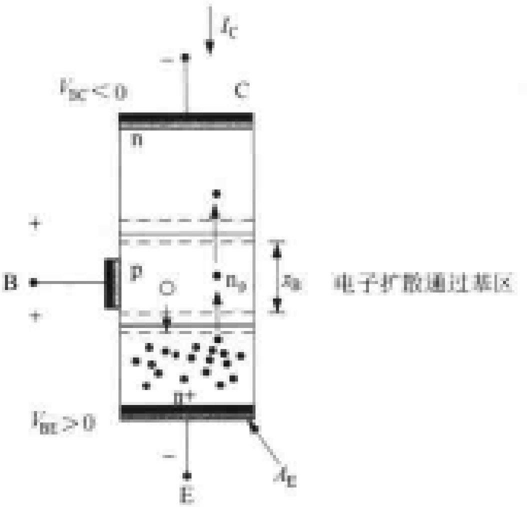


图 8.25　正向偏置工作情形

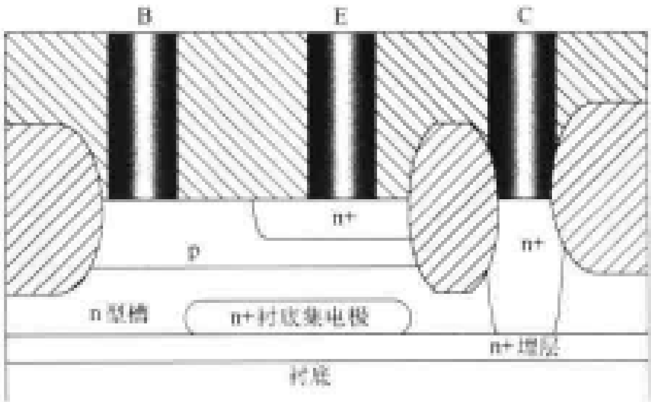


图 8.26　集成双极型晶体管门

BiCMOS 电路也采用截止和饱和的模型，它们概括在图 8.27 中。在截止时，两个 pn 结均为反向偏置，I_C 和 I_E 都近似为 0，如图 8.27(a)所示。当两个结都为正向偏置时，器件处于饱和状态，这一情形显示在图 8.27(b)中。此时，电流值由连接管的外部电路决定。两个 pn 结的电压值 $V_{BE(sat)}$和 $V_{BC(sat)}$为常数，典型值分别为 0.8 V 和 0.7 V 左右。因此根据基尔霍夫定律集电极-发射极的电压约为 $V_{CE(sat)} = 0.1$ V。

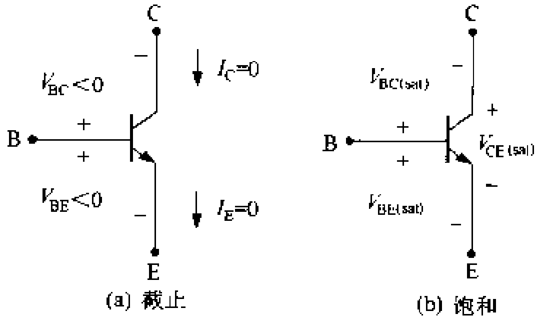


图 8.27 BJT 的截止和饱和情形

8.4.2 驱动电路

BiCMOS 电路中，CMOS 逻辑电路连至双极输出驱动器级。图 8.28 为一般的结构。CMOS 电路用来提供逻辑操作并驱动输出双极管 Q1 和 Q2。每次只有一个 BJT 是工作的。管 Q1 提供高输出电压，而 Q2 使输出电容放电达到低电平输出状态。

图 8.29 的反相电路是一个详细工作的例子。NOT 逻辑操作由 FET Mp 和 Mn 完成，尽管它们相互分开。其他两个 FET M1 和 M2 用来提供路径分别从 Q1 和 Q2 的基极移去电荷，这可加速该电路的切换，加强了它作为输出驱动器的用途。

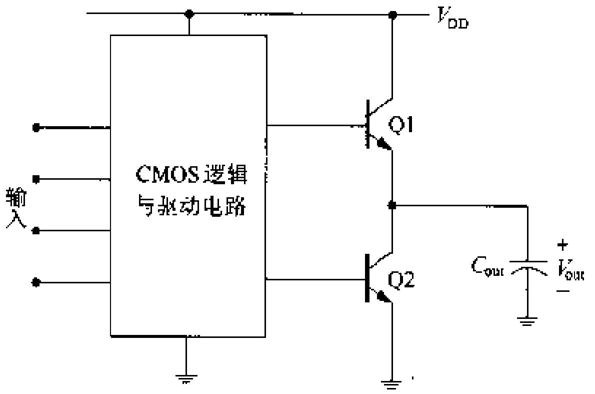


图 8.28 BiCMOS 电路的一般形式

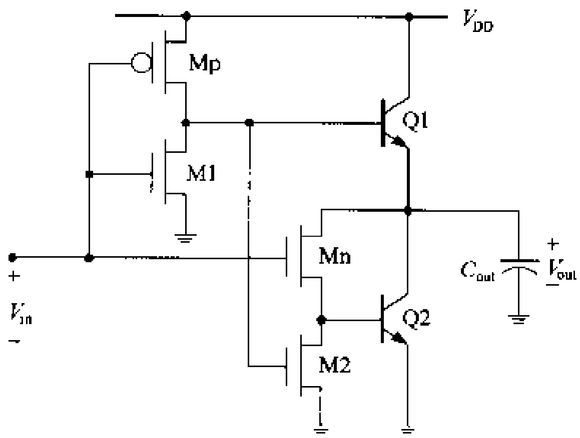


图 8.29 一个反相 BiCMOS 驱动电路

让我们来考察一下这个电路的 DC 工作。首先考虑输入电压值为 $V_{in}=0$ V 的情形。它使 Mp 导通，而 M1 和 Mn 截止。因为 Mp 和 M1 形成了一个反相器，Q1 的基极为高电压 V_{DD}，所以它变为导通；同一电压也使 M2 导通，它使 Q2 的基极接地从而驱动 Q2 至截止。这一情形的输出高电压 V_{OH} 可从图 8.30(a)所示的子电路计算出来。注意 Q1 将最终进入饱和状态。我们得到：

$$V_{OH} = V_{DD} - V_{BE(sat)} \qquad (8.193)$$

这是因为从基极到输出端电压降为 $V_{BE(sat)}$。$V_{in}=V_{DD}$ 情形时的子电路显示在图 8.30(b)中。现在看到 Mp 截止而 M1 和 Mn 导通。M1 使 Q1 的基极连至地从而驱动它至截止。这也使 M2 截止，所以输出电压送至 Q2 的基极而使 Q2 得到偏置。可以看到输出低电压为：

$$V_{OL} = V_{BE(sat)} \qquad (8.194)$$

这是因为 Q2 基极与发射极间有电压降。这一电路的问题是输出逻辑摆幅为 V_{DD} 减去 $2V_{BE(sat)}$。附加一些管子可以使这一问题缓解或消除。

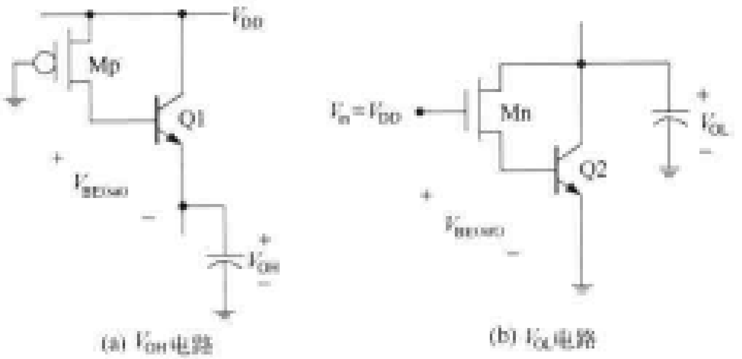


图 8.30　输出电压的 DC 分析

【例 8.7】 若 BiCMOS 电路的电源电压为 $V_{DD}=5\text{V}$。假设 $V_{BE(sat)}=0.8\text{ V}$，则

$$\begin{aligned} V_{OH} &= 5-0.8 = 4.3\ \text{V} \\ V_{OL} &= 0.8\ \text{V} \end{aligned} \tag{8.195}$$

这意味着在输出端的逻辑摆幅为 3.4 V。它可以通过重新设计输出级来改进。

以上的 CMOS 电路可以修改以提供所需要的逻辑功能。以这个设计为基础的 NAND2 门显示在图 8.31 中。对这个电路的仔细考察表明，这个逻辑是由驱动 Q1 的并联 pFET 以及在 Q2 集电极和基极之间的串联 nFET 构成的。另一些 FET 是下拉器件，用来使输出管截止。其他逻辑功能也可以以此为基础进行设计。一般地，上面输出管采用标准设计的 CMOS 电路作为驱动器，其中 nFET 部分重复出现在下面输出管的集电极和基极之间；再把下拉 nFET 加到基极则完成该设计。

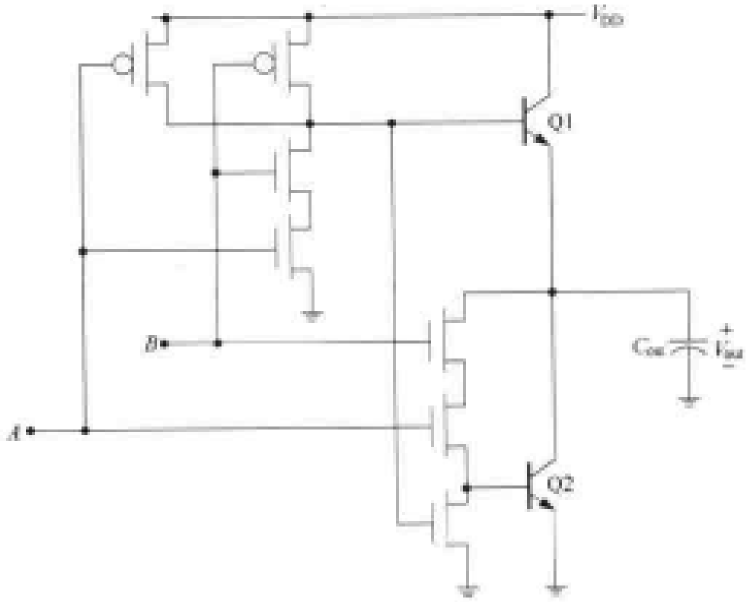


图 8.31　BiCMOS NAND2 电路

显然，BiCMOS 电路比起它们等效的 CMOS 电路来较复杂。如果写出总的输出电容：

$$C_{out} = C_{transistors} + C_L \tag{8.196}$$

式中，C_L 是外部负载，在 BiCMOS 电路中由于存在附加的器件，管子的寄生电容 $C_{transistor}$ 较大。由此可以得出一个重要的结论：BiCMOS 只对较大的 C_L 值比较有效。图 8.32 为一个典型的时间延时 t_d 与 C_L 的关系图。由于在 BiCMOS 中存在较高的器件寄生电容，所以 CMOS 和 BiCMOS 的特性曲线相交在 $C_L = C_X$ 处。对于 $C_L < C_X$，标准 CMOS 设计比BiCMOS电路提供更快的开关速度。只有当负载 C_L 比 C_X 大很多时才能看到 BiCMOS 速度的增加。这就使 BiCMOS 电路的应用限制在如驱动长的数据总线的情形中。而且在数字 VLSI 中运用这个技术时，BiCMOS 的成本及 V_{BE} 压降的问题也是一个需要考虑的重要因素。

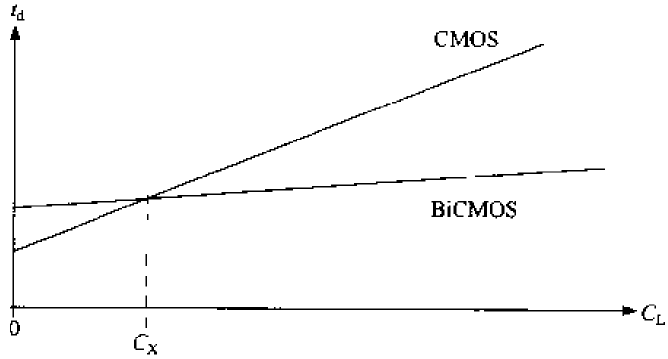


图 8.32 门延时与外部负载电容的关系

8.5 参考资料

[1] R. Jacob Baker, Harry W. Li, and David E. Boyce, **CMOS Circuit Design, Layout, and Simulation**, IEEE Press, Piscataway, NJ, 1998.

[2] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[3] Kerry Bernstein, et. al, **High Speed CMOS Design Styles**, Kluwer Academic Publishers, Norwell, MA, 1998.

[4] Ken Martin, **Digital Integrated Circuits**, Oxford University Press, New York, 2000.

[5] Robert F. Pierret, **Semiconductor Device Fundamentals**, Addison Wesley, Reading, MA, 1996.

[6] Jan M. Rabaey, **Digital Integrated Circuits**, Prentice Hall, Upper Saddle River, NJ, 1996.

[7] Jasprit Singh, **Semiconductor Devices**, John Wiley & Sons, New York, 2001.

[8] Ivan P. Sutherland, Bob Sproull, and David Harris, **Logical Effort**, Morgan-Kauffman Publishers, Inc., San Francisco, 1999.

[9] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[10] Neil H. E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, 1993.

[11] Edward S. Yang, **Microelectronic Devices**, McGraw-Hill, New York, 1988.

8.6 习题

[**8.1**] 一个 CMOS 反相器电路具有以下特性：

$$\begin{aligned} C_{\rm L} &= 100\ \text{fF} \qquad t_{\rm r} = 123.75\ \text{ps} \\ C_{\rm L} &= 115\ \text{fF} \qquad t_{\rm r} = 138.60\ \text{ps} \end{aligned} \tag{8.197}$$

反相器设计成对称的，$\beta_{\rm n}=\beta_{\rm p}$，且 $V_{\rm Tn}=|V_{\rm Tp}|$。

(a) 求 FET 的电阻 $R_{\rm n}=R_{\rm p}$ 且求 FET 的内部电容 C_{FET}。

(b) 求使这个电路 $t_f=t_r$ 的表达式。

(c) 若两个管子的宽度增加为它们原先值的 3.2 倍，求使 $t_f=t_r$ 的新的表达式，并且计算负载为 $C_{\rm F}=50$ fF 和 140 fF 时 $t_{\rm f}=t_{\rm r}$ 的值。

[**8.2**] 一个 CMOS 反相器的开关时间特性如下：

$$\begin{aligned} t_{\rm r} &= 430+3.68C_{\rm L}\ \text{ps} \\ t_{\rm f} &= 300+2.56C_{\rm L}\ \text{ps} \end{aligned} \tag{8.198}$$

式中，外部负载电容的单位为 fF。

(a) 对 $C_{\rm L}=0$ 至 $C_{\rm L}=200$ fF 的范围，画出上升和下降时间。

(b) 用相同的电路构成一个三级反相器链。如果每个 NOT 门的输出电容为 $C_{\rm L}=45$ fF，求最坏情形下通过该反相器链的延时。

[**8.3**] 考虑图 P8.1 的逻辑链。在 A 处的输入从 1 切换至 0。运用图 8.6 电路的设计步骤求出通过这个链延时的表达式。

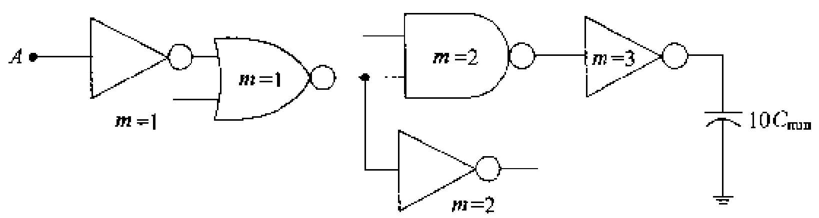


图 P8.1

[**8.4**] 一个 CMOS 工艺的特性参数为 $C_{\rm ox}=8\ \text{fF}/\mu\text{m}^2$，$r=2.6$ 及 $L=0.4\ \mu$m。nFET 和 pFET 的阈值电压数值相等。采用沟宽 2.2 μm 的 nFET 设计一个对称反相器。它作为在输出端驱动负载 $C_{\rm L}=38$ pF 的一个驱动器链的输入级。设计产生一个从输入级至负载延时最小的反相信号。

(a) 计算反相器的输入电容 $C_{\rm in}$，单位为 fF。

(b) 应用理想的尺寸放大原理求该链需要的级数。

(c) 已知沟宽 $W=1\ \mu$m nFET 的电阻为 $R_{\rm n}\approx 1725\Omega$，基于这个条件能否求出通过该链的总延迟时间？如果不能，那么还需要其他什么信息？

[**8.5**] 设计一个驱动器链驱动一个负载电容 $C_{\rm L}=40$ pF，假设最初一级的输入电容为 $C_{\rm in}=50$ fF。采用理想化的尺寸放大原理，决定级数及相应的尺寸。

[**8.6**] 一条互连线的单位长度电容为 $c' = 0.86$ fF/cm。这条线本身经过芯片的相当大部分,其总长度为 272 μm。一个"标准"反相器的输入电容为 52 fF 且用对称器件 $\beta_n = \beta_p$,该工艺的迁移率比为 $r = 2.8$。它用来作为该互连线驱动器链中的第一级。

采用理想理论设计这一驱动器链,并且要求输出必须是非反向的。

[**8.7**] 对 $\tau_x = 0.72\tau_r$ 的情形,求解方程(8.93)。

[**8.8**] 考虑图 P8.2 所示的逻辑链。采用逻辑努力技术,求出使该链延时最小所需要的每级的相对尺寸。假设为对称门,$r = 2.5$。

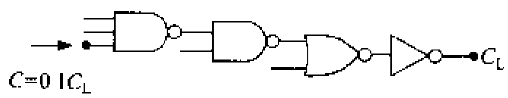


图 P8.2

[**8.9**] 图 P8.3 的逻辑链是以 $r = 2.5$ 的工艺制造的。采用逻辑努力技术,确定用粗黑线表示的路径中每级的最优尺寸。

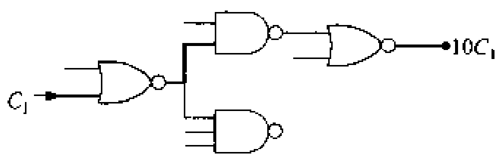


图 P8.3

[**8.10**] 考虑图 8.29 的 BiCMOS 反相器。假设在下面 BJT Q2 的位置上用一个较大的 nFET 代替,但 Q1 仍作为上拉驱动器不变。

画出所得到的只包括 Q1 所需要的 CMOS 驱动器电路的电路图。这个设计的逻辑摆幅是什么?

[**8.11**] 以图 8.29 的电路作为基础,设计一个 BiCMOS NOR2 电路。

[**8.12**] 设计实现如下功能的 BiCMOS 数字电路:

$$f = \overline{a + b \cdot c} \tag{8.199}$$

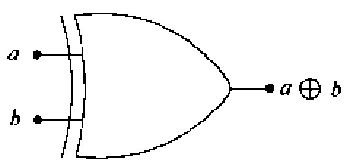
[**8.13**] 你能否保留已讨论过的 BiCMOS 电路的基本结构,但修改一下输出电路使 $V_{OH} = V_{DD}$ 及 $V_{OL} = 0$? 提示:回想一下一个标准的 CMOS 设计具有这些输出值。

第 9 章　CMOS 逻辑电路的高级技术

各种各样的 CMOS 电路设计风格已为大家熟知，它们在高速 VLSI 电路设计中非常有用。所有这些都是基于简单的逻辑门，但却以不同的方式工作。随着近年来 VLSI 应用的迅速增长，大多数高级技术已经开发出来，以克服在其中出现的一个或多个问题。有些问题是非常一般的，而另外一些则只用于特殊的情形。本章将有选择地解释一些运用在 VLSI 中的现代 CMOS 电路技术，将为以后几章的应用打下基础。

9.1　镜像电路

镜像电路基于串－并联逻辑门，但通常较快并具有较为一致的版图。一个镜像的基本概念可以从图 9.1 的 XOR 真值表中看出。输出为 0 意味着一个 nFET 链导通接地，而输出为 1 意味着一组 pFET 从电源得到了电流。这一观察的重要一面是产生 0 和 1 的输入组合数目相同。



a	b	$a \oplus b$	导通的器件
0	0	0	← nFET
0	1	1	← pFET
1	0	1	← pFET
1	1	0	← nFET

图 9.1　XOR 功能表

一个镜像电路对 nFET 和 pFET 采用相同的拓扑连接晶体管。这个概念应用于 XOR 功能，产生图 9.2(a)的电路。该图显示每个分支的输入组合。"镜像"效应可以通过把一面镜子沿输出线面向上或面向下放置来理解。在镜子中看到的将是该电路的另一边。图 9.2(b)为一个 XOR 单元的版图；pFET 的尺寸比 nFET 大以补偿它较小的工艺互导(k')值。

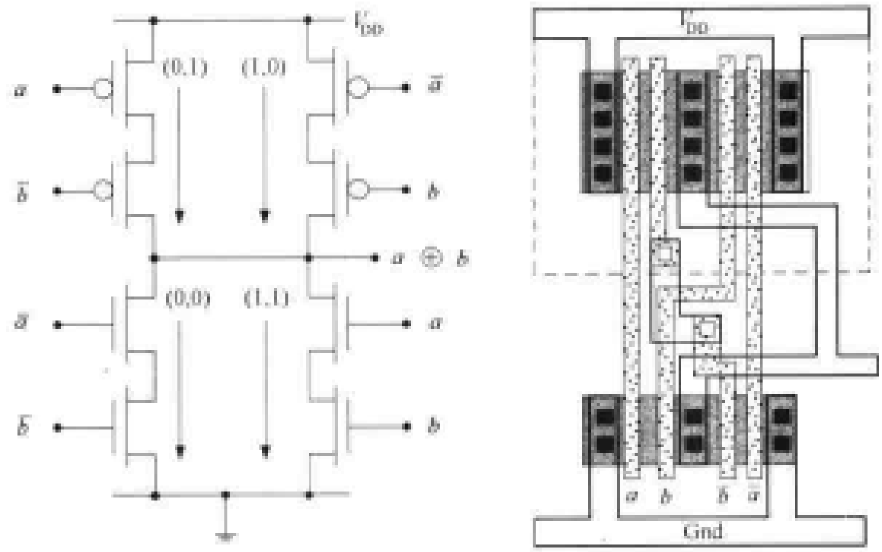
镜像电路的优点是有比较对称的版图以及较短的上升和下降时间。后一个优点可以通过图 9.3 的 RC 开关模型来理解。在输出和电源线或地线之间的每条路径由两个电阻和一个 FET 间的寄生电容构成。它的 Elmore 时间常数为：

$$\tau_x = C_{out}(2R_x) + C_x R_x \tag{9.1}$$

式中下标 x 或者是 n 或者是 p 取决于所指的路径。若把输出电压近似为指数变化则得到如下的上升和下降时间的表达式：

$$\begin{aligned} t_r &\approx 2.2\tau_p \\ t_f &\approx 2.2\tau_n \end{aligned} \tag{9.2}$$

虽然这个式子与 AOI 电路相同，但上升时间较小，因为寄生电容 C_p 较小。这是由于镜像电路只有两个 pFET 构成 C_p，而 AOI 电路在该节点上有 4 个晶体管。



(a) 电路　　(b) 版图

图 9.2　XOR 镜像电路

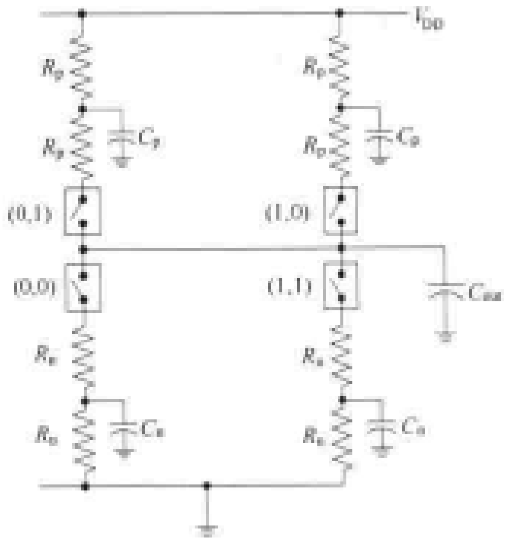
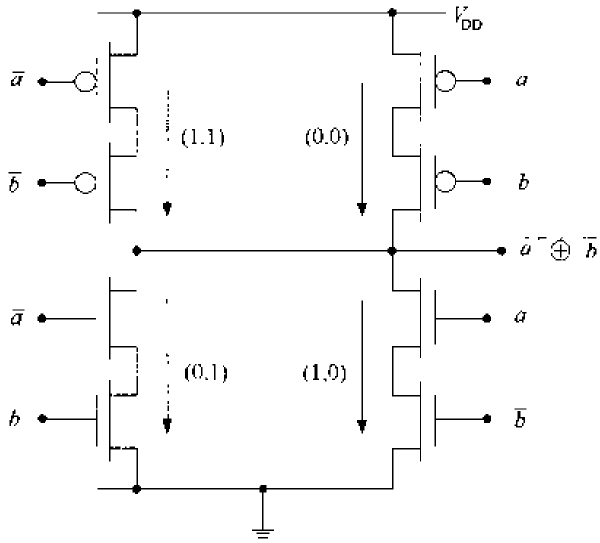


图 9.3　计算过渡过程的开关模型

这一概念可以很容易用来构成图 9.4 的 XNOR 电路。它具有与 XOR 相同的基本结构。以下的关系式

$$\overline{a \oplus b} = \overline{\bar{a} \cdot b + a \cdot \bar{b}} \tag{9.3}$$

表明只需要把输入 a 和 $\bar{a}$ 互换一下。其他镜像电路将在本书后面讨论特定应用，例如加法器电路时再介绍。



a	b	$\overline{a\oplus b}$
0	0	1
0	1	0
1	0	0
1	1	1

图 9.4 异或非(XNOR)镜像电路

9.2 准 nMOS 电路

在广泛采用 CMOS 之前，主要采用单个 FET 极性的逻辑电路。许多微处理器都曾用只有 nFET 的电路，即 nMOS 工艺来设计。尽管只有 nMOS 的电路由于大的 DC 功耗已被放弃，但一些主要的概念仍用在 CMOS 工艺中。把单个 pFET 加到其余都是 nFET 的电路产生一种逻辑系列，称为**准 nMOS**。

准 nMOS 逻辑采用较少的晶体管，因为只需用 nFET 逻辑块来产生逻辑功能。对于 N 个输入，一个准 nMOS 逻辑门要求 $(N+1)$ 个 FET。在通常的 CMOS 中，加入 pFET 组以减少 DC 功耗，但这一逻辑是多余的。标准的 N 个输入的 CMOS 门采用 $2N$ 个管子。

一个准 nMOS 门的基本拓扑连接画在图 9.5 中。单个的 pFET 被偏置成导通状态，因为它接地的栅极使 $V_{SGp}=V_{DD}$。它的作用是一个**上拉**器件，试图把输出 f 上拉到电源电压 V_{DD}。逻辑是由 nFET 阵列实现的，它可以采用已经学习的相同技术设计。这一阵列的作用如同一个在输出 f 和接地之间的大开关。如果开关开路，则 pFET 把输出上拉至电压 $V_{OH}=V_{DD}$。如果 nFET 开关闭合，则该阵列如同一个下拉器件，它力图把 f 下拉至地。然而由于 pFET 总是偏置在导通状态，所以 V_{OL} 将不可能达到理想的 0V 值。采用准 nMOS 电路的意图是减少 FET 的数目和面积。然而这一逻辑系列是比较复杂的，因为管子的相对尺寸决定了 V_{OL} 的数值，因此必须仔细使 V_{OL} 足够小以保证它是一个电气上为逻辑 0 的电压。

为了说明如何确定尺寸的问题，让我们分析一下如图 9.6 所示的简单反相器。输入电压设为 $V_{in}=V_{DD}$ 所以输出电压为 V_{OL}。电流 $I_{Dn}=I_{Dp}$。假设 V_{OL} 很小，那么 pFET 将饱和而 nFET 工作在非饱和区。于是 KCL(基尔霍夫电流)方程就有如下形式：

$$\frac{\beta_n}{2}[2(V_{DD}-V_{Tn})V_{OL}-V_{OL}^{2}] = \frac{\beta_p}{2}(V_{DD}-|V_{Tp}|)^2 \qquad (9.4)$$

这是一个关于 V_{OL} 的二次方程。求解这个方程得到如下有实际意义的根：

$$V_{OL}=(V_{DD}-V_{Tn})-\sqrt{(V_{DD}-V_{Tn})^2-\frac{\beta_p}{\beta_n}(V_{DD}-|V_{Tp}|)^2} \tag{9.5}$$

因此 V_{OL}的值取决于比率$(\beta_n/\beta_p)>1$。增加这一器件比率可以降低输出低电压。由于这一特点，准 nMOS 属于有比逻辑类型，即器件的相对尺寸决定 V_{OL}或 V_{OH}。

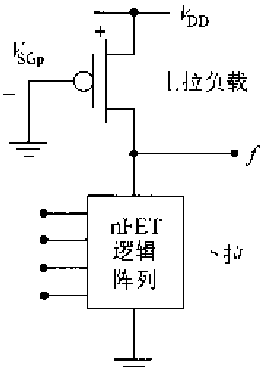


图 9.5 准 nMOS 逻辑门的一般结构

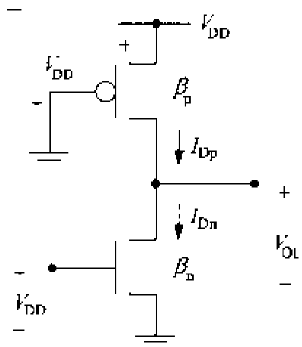


图 9.6 准 nMOS 反相器

【例 9.1】 考虑一个 CMOS 工艺，$V_{DD}=5$ V，$V_{Tn}=+0.7$ V，$V_{Tp}=-0.8$ V，$k'_n=150$ $\mu A/V^2$，及 $k'_p=68\ \mu A/V^2$。一个准 nMOS 反相器的尺寸为 $(W/L)_n=4$ 及$(W/L)_p=6$，于是这个反相器的输出电压为

$$V_{OL}=4.3-\sqrt{(4.3)^2-\frac{408}{600}(4.2)^2}=1.75\ \text{V} \tag{9.6}$$

这个输出低电压太大，因为它不会被同样类型的电路解释成逻辑 0。如果把 nFET 的尺寸增加到$(W/L)_n=8$ 且把 pFET 的尺寸减小到$(W/L)_p=2$，则由计算得到

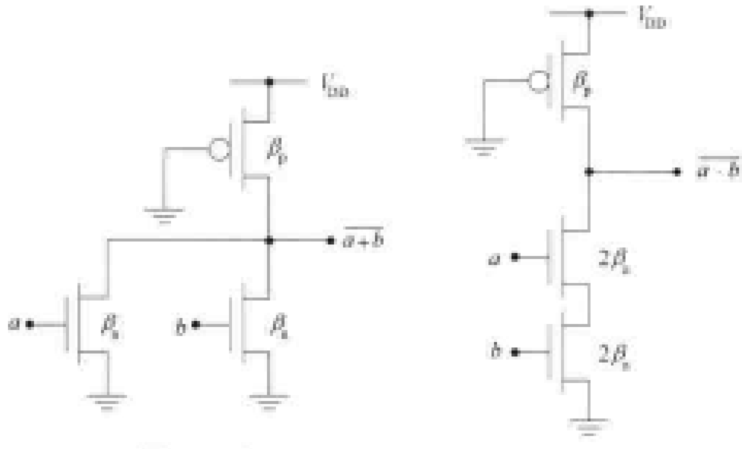
$$V_{OL}=4.3-\sqrt{(4.3)^2-\frac{136}{1200}(4.2)^2}=0.24\ \text{V} \tag{9.7}$$

这个值可以接受，因为它比使 nFET 导通的电压 $V_{in}=V_{Tn}$还小。这表明选择尺寸比，对这个设计类型是很关键的。应当注意当 $V_{in}=V_{DD}$时，从 V_{DD}到地形成了一条电流通路，从而引起很大的 DC 功耗。这也是限制准 nMOS 电路使用的另一个因素。

一般的准 nMOS 逻辑门是用与标准 CMOS 相同的 nFET 阵列设计的。图 9.7 为 NOR2 和 NAND2 的例子。让 β_n和 β_p 为一反相器的器件参数值。图 9.7(a)的 NOR2 门可以基于相同的 β 值，因为最坏情形下的下拉状态发生在只有一个 nFET 导通的时候。这个结论也可延伸到 N 个输入的 NOR 门中。图 9.7(b)的 NAND2 门因存在 nFET 串联而变得比较复杂。为了得到与反相器相同的下拉特性，实现逻辑的晶体管必须增大到 $2\beta_n$以使从输出到地的 nFET 电阻相同。这是要求串联逻辑 FET 的准 nMOS 逻辑门的一般问题。

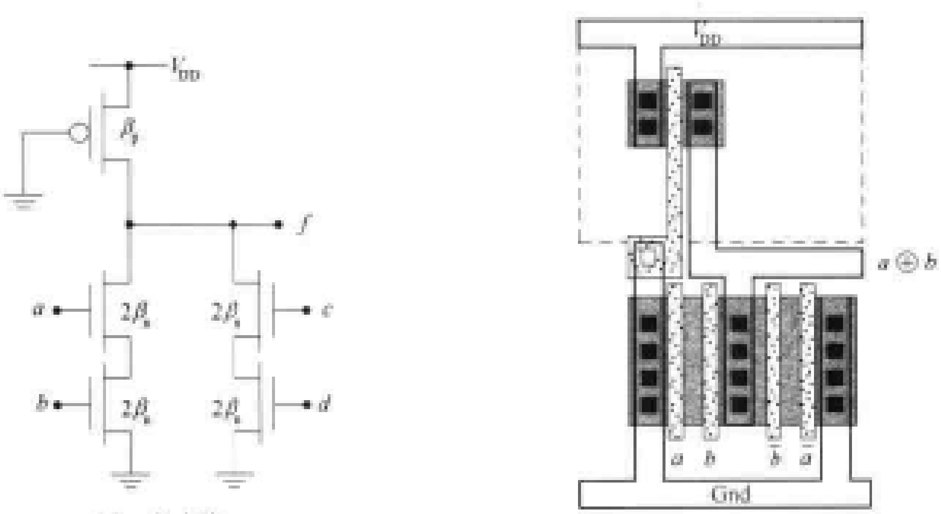
运用相同确定尺寸原理的一个基本 AOI 电路显示在图 9.8(a)中。产生较小较简单版图的优点可以从图 9.8(b)的 XOR 电路看到。因为只用单个 pFET，所以连线要简单得多。然而需要调整尺寸以保证与下一级有合适的电气耦合。准 nMOS 的有关问题使它的运用限制在

版图问题是关键的场合或某些特殊的开关电路场合，因为它可以产生较为简单的电路。



(a) NOR2门　　(b) NAND2门

图 9.7　准 nMOS NOR 和 NAND 门



(a) 一般电路　　(b) 版图例子

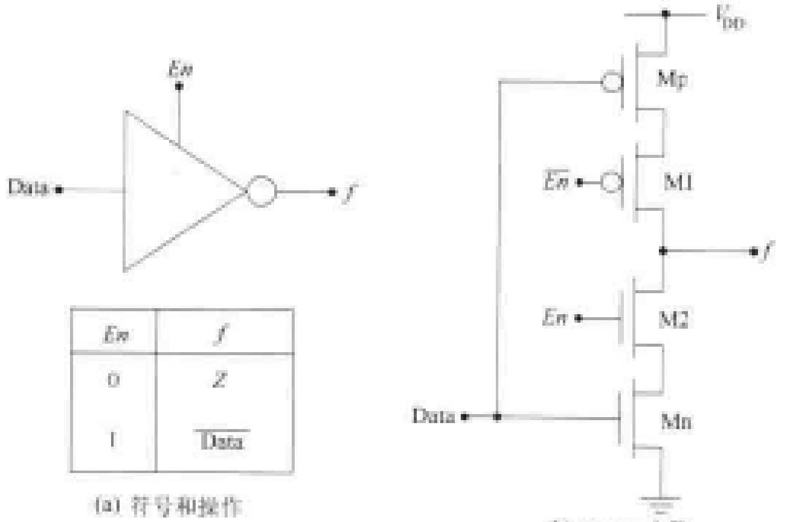
图 9.8　准 nMOS 逻辑的 AOI 门

9.3　三态电路

一个三态电路产生通常的 0 和 1 电压，同时也具有与开路相同的第 3 种状态即高阻 Z（或 Hi－Z）状态。三态电路对把电路与公共总线隔开是很有用的。

图 9.9(a)为三态反相器的符号图。使能信号 En 控制它的工作。当 $En=0$，输出处于"三态"，这意味着 $f=Z$。正常工作发生在当 $En=1$ 时。图 9.9(a)为一个 CMOS 电路。FET M1 和 M2 是三态器件。$\overline{En}$ 信号应用到 pFET M1，而 En 则控制 M2。当 En ＝ 0 时，M1 和 M2 都截止，此时输出同时与电源线和地线隔离。这就是 Hi－Z 状态的电路情况。注意，即使没有任何硬布线连接，输出电容（这里没有明显表示出来）也可维持一个电压。当 $En=1$ 时，M1 和 M2 同时导通，于是 Mp 和 Mn 的作用如同一个由 Data 控制的反相器。三态反相器的版图很容易画出，如图 9.10 所示。

在输入端增加一个通常的静态反相器可以得到一个非反相的电路(缓冲器)。由于它们的广泛应用,单元库中通常包括几种反相和非反相的三态电路。



En	f
0	Z
1	$\overline{Data}$

(a) 符号和操作　　(b) CMOS电路

图 9.9　三态反相器

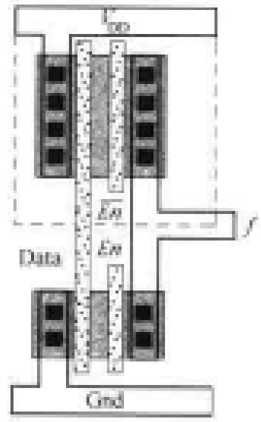


图 9.10　三态反相器的版图

9.4　时钟控制 CMOS

至此,考察的所有电路从本质上讲是完全**静态**的。只要一个静态逻辑门的输入值是合法的且电路稳定,那么它的输出就是合法的。逻辑延时的产生是由于信号波动通过电路,它们也没有参照任何规定的时间基准。只有发展到想用时钟控制和时序电路的时候,数字逻辑的真正功能才能实现。在本节中,将考察称为**时钟控制 CMOS** 的基本设计类型,或简称为 C^2 MOS。

时钟信号 ϕ(或 Clk)是一个周期波形,它有明确定义的周期 T[秒]和频率 f[Hz]即:

$$f = \frac{1}{T} \tag{9.8}$$

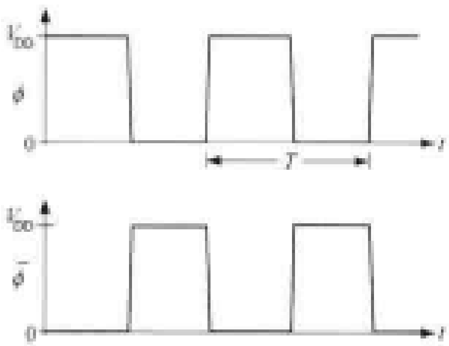


图 9.11　时钟信号

图 9.11 表示时钟 $\phi(t)$及它的反信号 $\bar{\phi}(t)$。理想上,它们是**不互相重叠**的,即对所有时间 t

$$\phi(t) \cdot \bar{\phi}(t) = 0 \tag{9.9}$$

然而如果 $\phi(t)$定义为具有最小值 0V 和最大值 V_{DD},那么

$$\bar{\phi}(t) = V_{DD} - \phi(t) \tag{9.10}$$

所以在过渡期间,时钟有稍微的重叠。因此建立起一组在所有时间都完全不重叠的时钟也许是很有用的。

图 9.12 为 C^2MOS 门的一般结构。它包括一个静态逻辑电路及由 ϕ 和 $\bar{\phi}$ 控制的三态输出(由 FET M1 和 M2 组成)。这个电路的工作可以通过所显示的时钟波形来理解。当 $\phi = 1$ 时,M1 和 M2 同时导通。因为 pFET 和 nFET 逻辑块同时连到了输出节点,所以此时电路就简化成一个标准的静态逻辑门。在这期间,输出 $f(a,b,c)$有效,在输出电容 C_{out}上建立了电压

V_{out}。当时钟变为 $\phi=0$ 值时，M1 和 M2 同时处于截止状态，所以输出端处在一个高阻状态 Hi－Z。在这段时间，FET 逻辑阵列没有连接到输出端，所以输入对该电路没有任何影响，此时输出电压维持在 C_{out}上直到时钟返回到 $\phi=1$ 值。

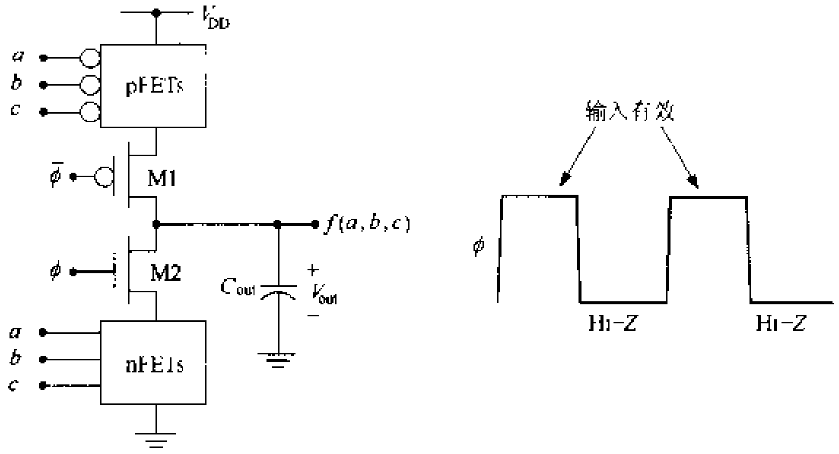


图 9.12 C^2MOS 门的结构

晶体管阵列的设计采用与标准逻辑门相同的技术。图 9.13 为 NAND 和 NOR 电路，分别画在图(a)和图(b)中。版图类似于三态电路，但用时钟代替使能信号。图 9.14 的版图提供了一种方法来放置和连接晶体管。注意存在串联的时钟 FET 自然会同时加大电路的上升和下降时间。

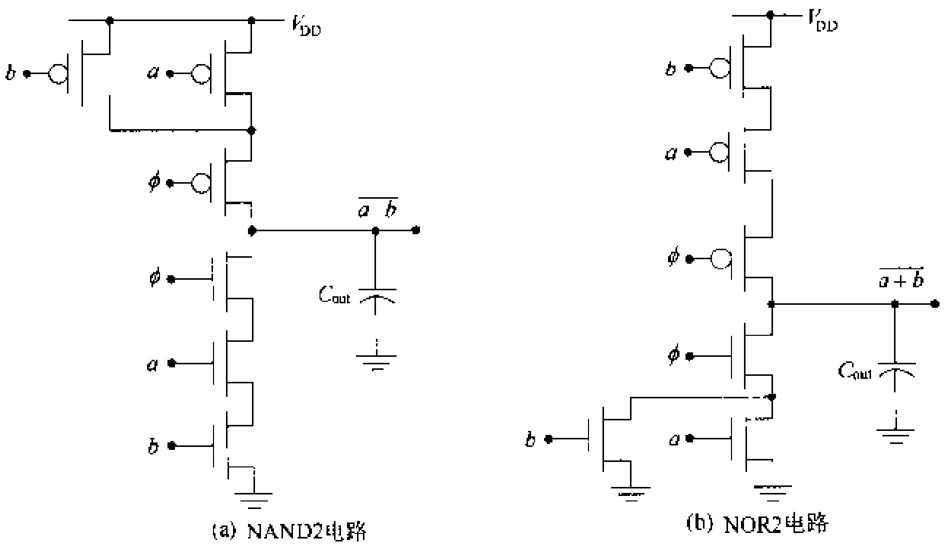


图 9.13 钟控 CMOS 逻辑门的例子

钟控 CMOS 非常有用，因为可以通过控制这个门的内部操作，同步通过一个逻辑链的数据流。ϕ 的每个周期允许一组新的数据进入这个电路中。一个缺点是输出节点不能长时间保持 V_{out}上的电荷，这是由于称为**电荷泄漏**的现象所致。它确定了所允许时钟频率的下限。

图 9.15 (a)显示了基本电荷泄漏现象。尽管晶体管都处在截止状态，但采用 FET 不可能阻止所有的电流。如果电压加在漏端或源端，就会有一个小的**漏电电流**流入或流出器件。漏电电流有许多来源。一个是由于如图所示需要与基底连接。pFET 的基底是 n 阱区域，它连接到电源电压 V_{DD}上。因为 pFET 的源端是 p+ 区，这就形成了一个 pn 结（一个二极管），它

允许一个小的漏电电流 i_p 流入该节点。nFET 也有相同的问题，即 i_n 从输出端流到 p 衬底。若用 i_{out} 表示从电容流出的电流，可以求这两个漏电电流的代数和得到：

$$\begin{aligned} i_{out} &= i_n - i_p \\ &= -C_{out}\frac{dV}{dt} \end{aligned} \tag{9.11}$$

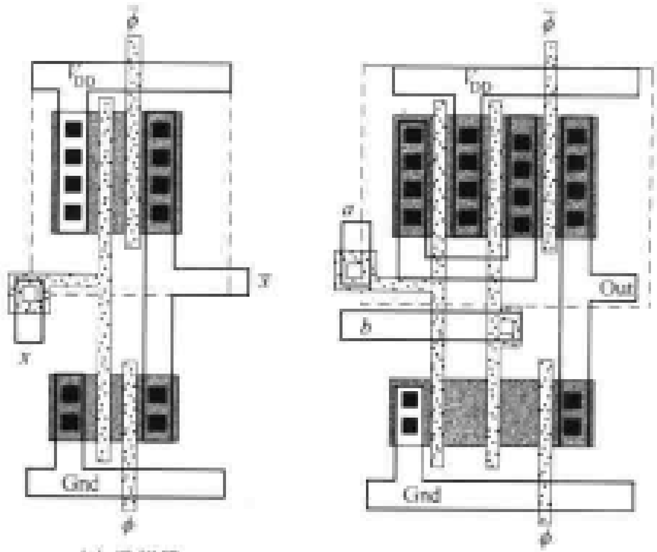


图 9.14 C² MOS 电路的版图例子

上式第二行中运用电容的 I-V 关系；注意式中的负号表明 i_{out} 是从正端流出的。

为了理解这个漏电电流的影响，假设在电容上存有初始电压 $V(t=0)=V_1$。如果 $i_n > i_p$，那么 $i_{out}=I_L$ 是一个正数，表明电流流出电容，该方程可重写为：

$$I_L = -C_{out}\frac{dV}{dt} \tag{9.12}$$

重新整理它得到

$$\int_{V_1}^{V(t)} dV = -\int_0^t \left(\frac{I_L}{C_{out}}\right) dt \tag{9.13}$$

假设 I_L 为常数，则将该方程积分得到

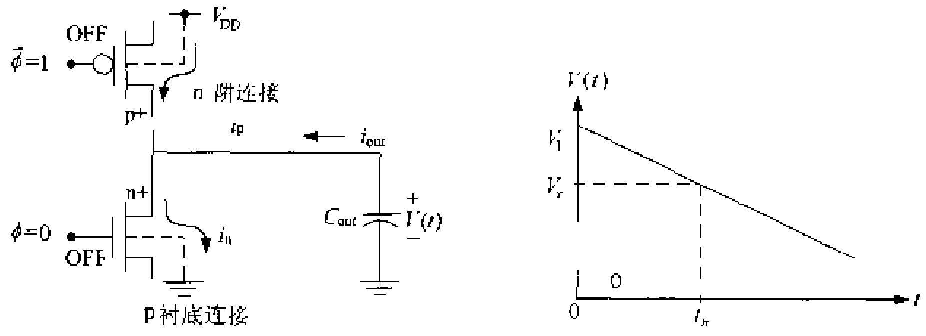
$$V(t) = V_1 - \left(\frac{I_L}{C_{out}}\right)t \tag{9.14}$$

这表明电压随时间线性衰减，如图 9.15(b)所示。随着电压减小，它最终达到了最小允许的逻辑 1 值如在图中所示的 V_x。如果 V 下降到这个值以下，它将被错误地识别为逻辑 0 电压。**维持时间** t_h 相应于逻辑 1 电压可以被存放的最长时间。按照定义，这发生在当

$$V(t_h) = V_1 - \left(\frac{I_L}{C_{out}}\right)t_h = V_x \tag{9.15}$$

重新整得到：

$$t_h = \left(\frac{C_{out}}{I_L}\right)(V_1 - V_x) \tag{9.16}$$



(a) 体泄漏电流 (b) 逻辑 1 的电压衰减

图 9.15 电荷泄漏问题

于是得到了这一情形的维持时间。为了估计维持时间的数量级,可把电容估计为 50 fF,漏电电流为 0.1 pA,而电压变化为 1V。由这些值得到

$$t_h = \left(\frac{50\times10^{-15}}{10^{-13}}\right)(1) = 0.5\ \text{s} \tag{9.17}$$

这对于宏观世界来说是一个非常短的时间。然而就现代数字 CMOS 的微观时间尺度而言,$t_h = 500$ ms 似乎像是无穷大!快速的时钟因此帮助我们避免了这个问题。这一估计确实表明不可能停止 C^2MOS 电路中的时钟信号。

如果 $V(t=0)=0$V 会发生什么?这相当于存放逻辑 0 电压。如果 $I_L = i_p - i_n > 0$,那么相同的分析成立,于是得到

$$V(t) = \left(\frac{I_L}{C_{out}}\right)t \tag{9.18}$$

也就是充电电流 I_C 使电压随时间增加。这意味着逻辑 0 电压也会变化,所以仍然对最小的时钟频率有要求。

在亚微米器件中,由于存在另一种 FET 漏电电流,即所谓**亚阈值电流** I_{sub} 而使电荷泄漏问题更加严重。亚阈值电流是甚至栅电压小于 V_T 时的漏-源电流。对亚阈值电流的简单估计为:

$$I = I_{D0}\left(\frac{W}{L}\right)e^{-(V_{GS}-V_T)/(nV_{th})} \tag{9.19}$$

式中 I_{D0} 随 V_{DS} 而变化。V_{th} 是温度在 300 K 时的热电势(KT/q)≈26 mV,而 n 是随电容而变化的参数。对 I_{D0} 的一个保守估计约为 10^{-9}A,它显著地减少了维持时间。对于前面的电容值和电压以及 $V_{GS}=0$,通过宽长比为 1 的 FET 的漏电维持时间估计为

$$t_h = \left(\frac{50\times10^{-15}}{10^{-9}}\right)(1) = 50\ \mu\text{s} \tag{9.20}$$

其他漏电电流来源于物理结构以及用来构成硅电路的材料。在亚微米器件中总的电荷泄漏电

流达到 $I_L = 0.1\ \mu A = 10^{-7} A$ 并不是不合理的。对于这个大小的漏电电流，维持时间将减小到

$$t_h = \left(\frac{50\times10^{-15}}{10^{-7}}\right)(1) = 0.5\ \mu s \tag{9.21}$$

这清楚地表明，在电容节点上电荷只能存储一个有限的时间，这对逻辑电路提出了重要的约束条件。

尽管为了简单起见，把漏电电流近似为常数值，一个深入的分析表明它们与电压有关。一般的微分方程有如下的形式：

$$I_L(V) = -C_{out}(V)\frac{dV}{dt} \tag{9.22}$$

式中注意到输出电容 C_{out} 也取决于电压。如果已知 $I_L(V)$ 和 $C_{out}(V)$ 的显函数，那么

$$\int_0^t dt = \int_{V_x}^{V(t)} \frac{C_{out}(V)}{I_L(V)}\, dV = t \tag{9.23}$$

即可以通过积分求得 $V(t)$。一种更为实际的方法是运用数值解法。这些量与 V 的关系引起一个非线性衰减，如图 9.16 所示的例子。维持时间仍然以相同的方式定义。在电路设计层次上，电荷泄漏的信息通常通过电路模拟得到。

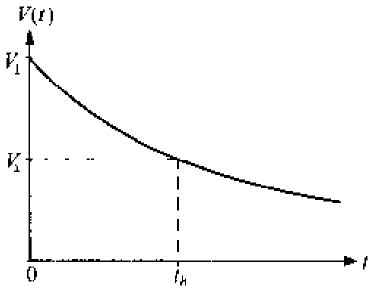


图 9.16 电压衰减的一般情形

电荷泄漏发生在每当试图用截止状态的 MOSFET 把电荷维持在一个节点电容上的时候。在本章其余部分的许多复杂电路都具有这一特点，因此应当记住需要检查这个问题。MOSFET 的简单 SPICE 模型并没有精确地考虑漏电电流，至今最好的结果是采用 BSIM 方程得到的。

进一步研究的目的

虽然电荷泄漏是动态电路中的一个重要问题，这一部分的讨论集中在运用 MOSFET 来实现"开路的开关"问题。随着尺寸缩小，漏至源的漏电电流增加，器件看上去越来越不像曾用来设计 CMOS 逻辑电路的理想开关。这是数字亚微米 VLSI 中最关键的问题之一。器件研究者们还在继续考察这个问题。就硅工艺而言，有两种主要的方法流行。

一种技术是通过运用不同的材料以及不同的 FET 结构，改进制造工艺来减少漏电。几年来，已经出现漏电电流大小可以控制的较好器件，电路设计者必须用这些器件进行设计。

另一个方法是开发新类型的晶体管代替标准的 MOSFET。已经提议和制造了具有改进特性的新器件，并且在文献中出现许多很有希望的结构。然而器件研究往往在最初只考虑构成单个的管子，而不是高密度的 VLSI 芯片。制造中的问题常常限制了在这些应用中使用这些器件。另一个问题是电路和逻辑设计者们必须先学习一种器件的特性，然后才能开发出数字设计方法学。对标准 MOSFET 适用的一种技术即使对基于具有不同 I-V 特性晶体管的电路能完全适用，也可能并不是最好的选择。

把 MOSFET 的尺寸缩小常被认为是工艺技术的自然进化。亚微米尺寸 FET 的开发对电路设计技术已经产生明显的影响。引入新的开关器件必定影响 VLSI 层次化设计中的所有层次，并且在高密度设计实现之前还必须完成许多研究。VLSI 设计者必须继续了解这个领域

的变化。

9.5 动态CMOS逻辑电路

动态逻辑门运用时钟和MOSFET的电荷存储特性,实现逻辑操作。时钟提供同步的数据流,它使这个技术在设计时序电路时十分有用。动态逻辑门的特点是计算结果只在一个短时间内有效。虽然这使电路较难设计和应用,但它们要求较少的晶体管并且可比静态逻辑链快。

动态电路基于图9.17所示的电路。时钟 ϕ 驱动一对互补管子Mn和Mp;它们控制该电路的工作并且实现同步。逻辑功能是由在输出节点和地之间的nFET阵列来实现的。输出电压 V_{out} 取自输出电容 C_{out} 两端的电压。

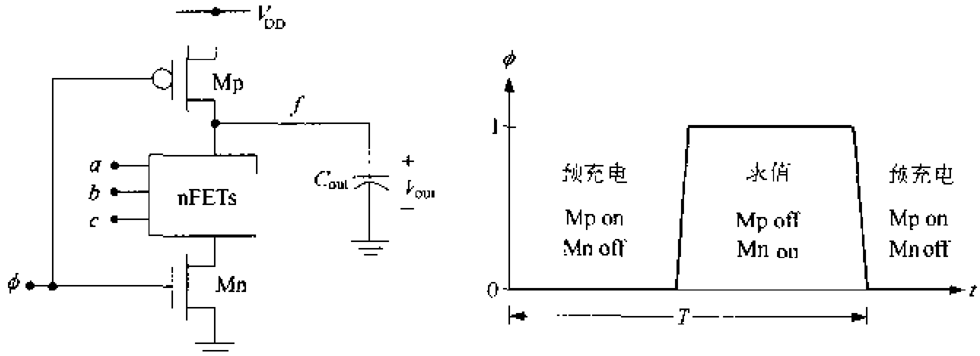


图9.17 基本的动态逻辑门

时钟信号 ϕ 在每个周期定义两个不同的工作模式。当 $\phi=0$ 时电路**预充电**,Mp导通而Mn截止。这在 V_{DD} 和输出之间建立了一条导电通路,使 C_{out} 充电到电压 $V_{out}=V_{DD}$。Mp通常称为预充电FET。由于预充电期间底部的nFET逻辑块并不连接到地,所以输入没有产生任何影响。

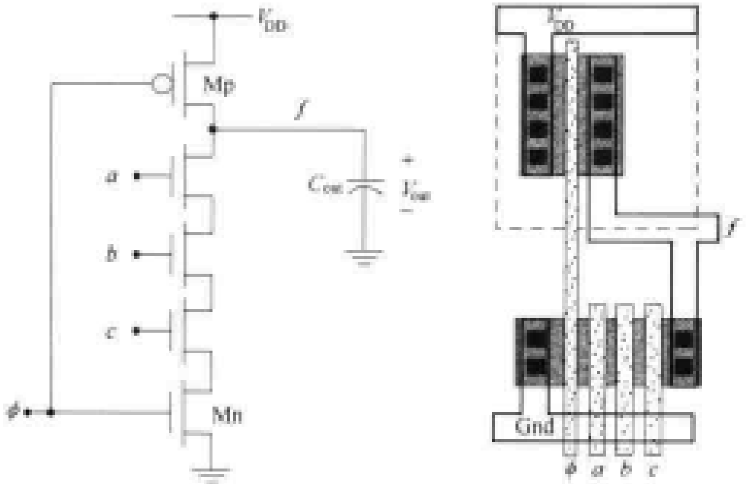
时钟过渡到 $\phi=1$ 则使这一电路驱动到**求值**模式,此时Mp截止而Mn导通,输入有效并且控制nFET逻辑阵列的切换;Mn通常称为求值管。如果这个逻辑块的作用像一个闭合的开关,那么 C_{out} 可以通过逻辑阵列和Mn放电,使最终结果 $V_{out}=0V$,这相应于逻辑 $f=0$。如果输入使这个逻辑块的行为如同一个从上到下开断的开关,那么在 C_{out} 上的电荷就保持着,所以 $V_{out}=V_{DD}$;逻辑上讲,这就是输出 $f=1$。电荷泄漏最终会使输出下降至 $V_{out}\to 0V$,这将是一个不正确的逻辑值。维持时间是由电路决定的,一般地,这个时间确定了对时钟最小频率的限制。

图9.18(a)为一个动态NAND3电路,采用三个串联的FET可以形成这个逻辑。其输出

$$f=\overline{a\cdot b\cdot c} \tag{9.24}$$

只是在 $\phi=1$ 的求值期间有效。版图很容易设计,如图9.18(b)的例子所示。由于求值nFET Mn与逻辑块串联,C_{out} 必须通过四个管子放电。加大nFET的尺寸将减少下降时间。

如在上面提及的,电荷泄漏减少了 $f=1$ 时在输出节点上维持的电压。对这个电路的详细分析表明,当时钟过渡到 $\phi\to 1$ 时,会发生另一个称为**电荷分享**的问题。它甚至在电荷泄漏变得显著之前,就会产生减少输出电压的影响。



(a) 电路　　(b) 版图

图 9.18　动态逻辑门例子

电荷分享问题来源于在 FET 之间的节点寄生电容 C_1 和 C_2，如图 9.19 所示。当时钟置为 $\phi=1$ 时，Mp 截止，使输出节点与电源隔离。在求值期开始时 C_{out} 上的初始电压为 $V_{out}=V_{DD}$。假设此时电容上的电压 V_1 和 V_2 均为 0 V，电路中的总电荷为

$$Q = C_{out}V_{DD} \qquad (9.25)$$

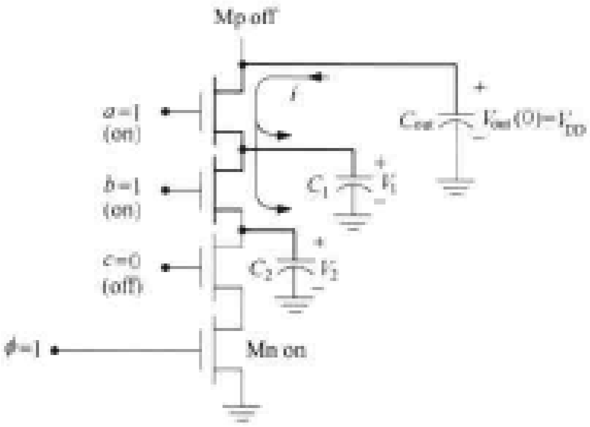


图 9.19　电荷分享电路

这个电路最坏情形下的电荷分享情形是当输入为 $(a,b,c)=(1,1,0)$ 时。当 $c=0$ 时，没有任何到地的放电路径，所以输出电压应当维持在高电平。然而因为输入 a 和 b 的 FET 导通，所以 C_{out} 在电气上连至 C_1 和 C_2，如粗黑线所示。由于 V_{out} 最初大于 C_1 和 C_2，所以就会有电流流动。这相应于电荷从 C_{out} 转移到 C_1 和 C_2。由关系 $Q=CV$ 可知，V_{out} 减小而 V_1 和 V_2 增加。当它们的电压相等时电流就停止流动，最终电压值为：

$$V_{out} = V_2 = V_1 = V_f \qquad (9.26)$$

在电路中的总电荷于是按下式分布

$$\begin{aligned} Q &= C_{out}V_f + C_1V_f + C_2V_f \\ &= (C_{out} + C_1 + C_2)V_f \end{aligned} \qquad (9.27)$$

由电荷守恒原理，它必定等于系统中最初的电荷：

$$Q = (C_{out} + C_1 + C_2)V_f = C_{out}V_{DD} \qquad (9.28)$$

求解最终电压得到

$$V_f = \left(\frac{C_{out}}{C_{out} + C_1 + C_2}\right)V_{DD} \qquad (9.29)$$

因为

$$\left(\frac{C_{out}}{C_{out}+C_1+C_2}\right)<1 \tag{9.30}$$

我们看到

$$V_f < V_{DD} \tag{9.31}$$

因此电荷分享减少了输出节点上的电压。为了保持 V_{out}较高，电容必须满足如下关系：

$$C_{out} \gg C_1 + C_2 \tag{9.32}$$

这也许较难实现，因为电容值是由版图尺寸决定的。在发生电容分享后，节点仍然会有电荷泄漏，它还会使电压继续随时间降低。

9.5.1 多米诺逻辑

多米诺逻辑是一种 CMOS 逻辑类型，它是将一个静态反相器加到基本动态门电路的输出端构成的。所形成的结构显示在图 9.20 中。预充电和求值过程仍然发生，但现在只是在动态级和反相器之间的 C_X 受到影响。时钟值 $\phi=0$ 定义了预充电。在这期间 C_X 被充电至电压 $V_X=V_{DD}$，它使输出电压 $V_{out}=0V$。输入在 $\phi=1$ 的求值期间有效。如果 C_X 保持住它的电荷，那么 V_X 维持高电平并且 $V_{out}=0V$，表明输出为逻辑 0。如果 C_X 放电，那么 $V_X\to0V$ 及 $V_{out}\to V_{DD}$，这相应于输出为逻辑 1。

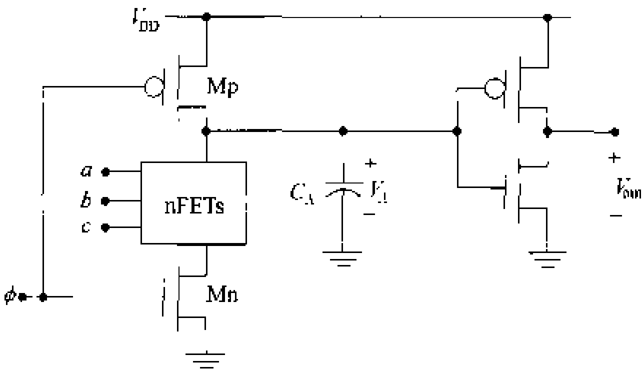


图 9.20 多米诺逻辑级

多米诺逻辑门由于它存在输出反相器，所以是**非反相**的。这一特性的两个例子显示在图 9.21 中。图 9.21 (a)中的 AND 门很容易理解：如果 $a=b=1$，那么内部节点放电至 0V，使输出为逻辑 1(V_{DD})。类似地，图 9.21 (b)中的 OR 门，无论 $a=1$ 或者 $b=1$ 都得到输出为 1。这使仅用多米诺门进行逻辑设计有些困难，因为一个完全的逻辑操作集要求有 NOT 操作。[①] 虽然我们可以增加反相器，但会发现它有可能把电路产生的假信号(毛刺)引入该电路中，而通常这是需要避免的。反相器只用在多米诺链的开始和末端处。图 9.22 为多米诺 AND3 门版图的例子。这实际上不过是一个动态 NAND3 电路串接一个静态反相器，所以版图保持了一般动态逻辑的特点。

多米诺逻辑是由于它的串联操作方式得名的。图 9.23 为一个三级电路。每级都是由相同的时钟相位 ϕ 来控制。在 $\phi=0$ 预充电期间，电容 C_1, C_2，和 C_3 同时被充电到 V_{DD}。这使输出 f_1, f_2 和 f_3 同时为 0。当 $\phi=1$ 时，整个链处于求值。在多米诺链中，这一点很像“多米诺链式反应”，即它必定开始于第 1 级，然后一级一级地传至输出端。为了理解这一点，假设第二级的输出 f_2 并且观察它在求值期间从它预充电值 $f_2=0$ 切换到 $f_2=1$。这一点能够发生

① 一个完全的逻辑操作集能产生任何逻辑组合。若没有 NOT 算子，像 XOR 和 XNOR 这样的功能就不可能实现。

的惟一方式是 C_2 放电，但这要求 $f_1=1$ 以使在放电链中的 nFET 导通。对第 1 级应用同样的理由，可知只有 C_1 放电 f_1 才能切换到 1。继续这一理由，可以看到 $f_3\to 1$ 只能发生在第 1 级和第 2 级都已发生了相同的过渡之后。

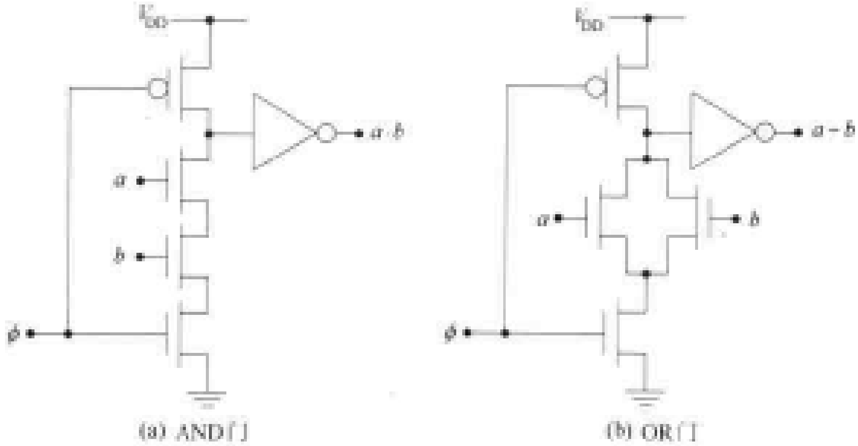


图 9.21 非反相多米诺逻辑门

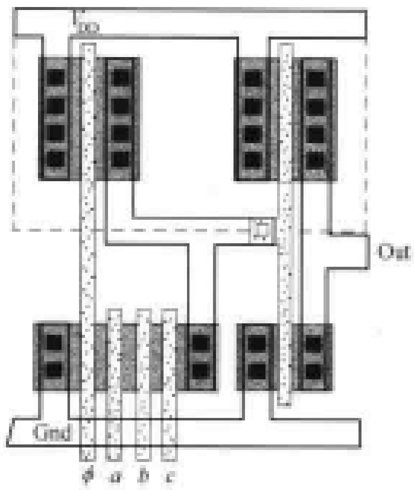


图 9.22 多米诺 AND门的版图

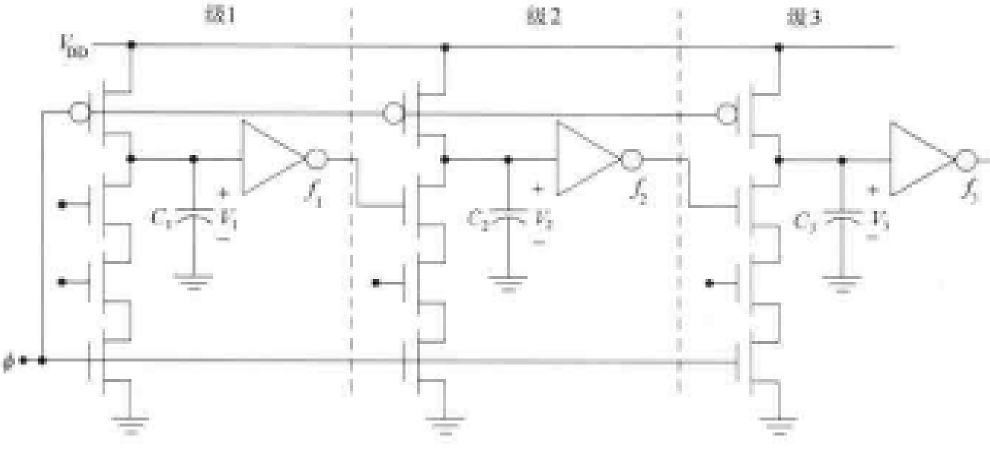
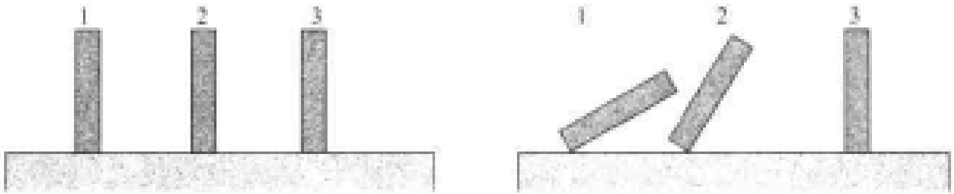


图 9.23 多米诺链

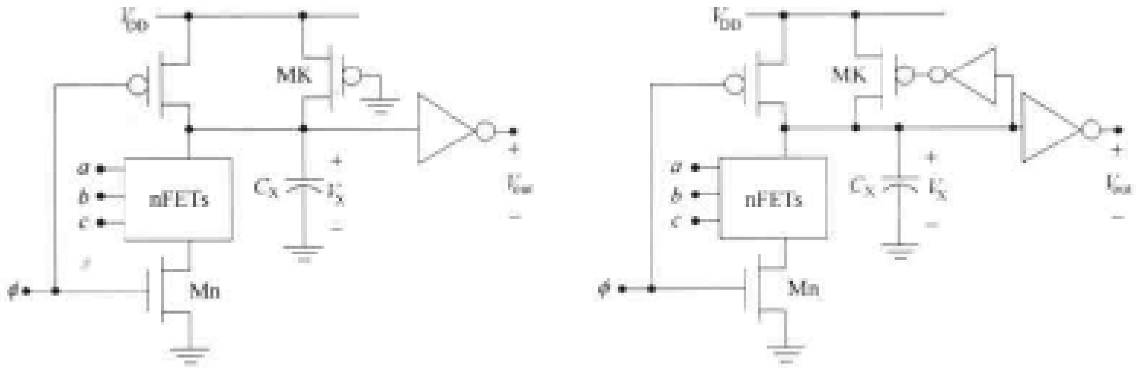
多米诺效应画在图 9.24 中以帮助理解这个过程。图 9.24(a)用直立的多米诺骨牌代表预充电过程。这个链的求值显示在图 9.24(b)中,使输出 $f\to 1$ 的放电过程用倒下的多米诺骨牌表示。它可以推倒下一级,但其他输入有可能不发生放电。在图中第 1 级和第 2 级已在放电,但第 3 级仍保持高电平(仍处在它预充电时的状态)。注意这个操作表明多米诺逻辑门只在串联链中有用。



(a) 预充电　　(b) 求值

图 9.24　多米诺效应示意

多米诺链必须有一个足够长的求值期间,以允许每级有时间放电。这意味着使中间电压 V_X 减小的电荷分享和电荷泄漏过程可能成为一个限制因素。已经开发**电荷保持电路**来克服这一问题。图 9.25 显示两种电路。在图 9.25(a)中,一个 pFET MK 被置于导通以允许一个小电流补充 C_X 上的电荷。电荷保持 FET 的宽长比应当很小,所以它不会明显地影响放电过程,这称为"弱"器件。另一种方法显示在图 9.25(b)中,一个反相器控制弱 pFET 的栅极。如果内部的 C_X 放电确实发生,那么输出电压 V_{out} 就会增加,通过这个反相器送到 pFET 使之截止,允许放电继续下去。



(a) 单个FET的电荷保持电路　　(b) 反馈控制的保持电路

图 9.25　电荷保持电路

基本多米诺电路的一个有趣的延伸是**多输出多米诺逻辑**(**MODL**)。这类电路使单个的逻辑门有两个或多个输出,这使它很有特点。图 9.26 为一个两输出的 MODL 结构。逻辑阵列被划分成两个不同的逻辑块,分别表示为 F 和 G,它们形成了一个附加的输出节点。增加一个反相器和一个预充电管就构成了两个输出:

$$\begin{aligned} f_1 &= G \\ f_2 &= F \cdot G \end{aligned} \tag{9.33}$$

通过研究这个逻辑电路很容易理解这一点。如果 G 逻辑块的作用如同一个闭合的开关,那么它产生输出 $f_1 = G$。如果这个情况发生,那么也可以使第 2 个逻辑块 F 的作用如同一

个闭合的开关而引起放电。这个 AND 操作的嵌套虽然非常有限，但它确实应用在几个重要的算法，如超前进位加法器中。

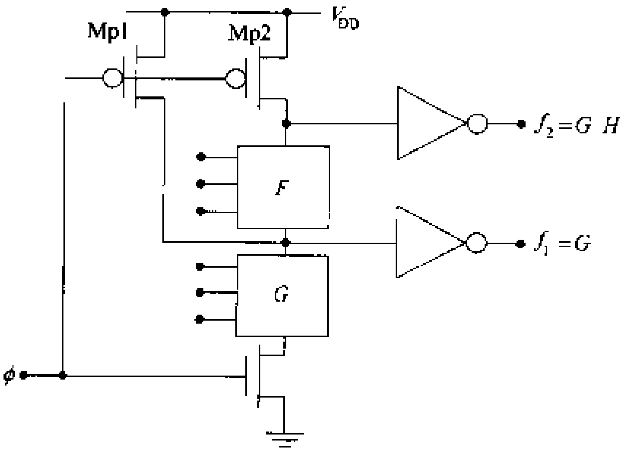


图 9.26 MODL 电路结构

9.5.2 动态逻辑电路的功耗

CMOS 动态逻辑电路可以用来设计非常快速的开关电路而只占有中等程度的芯片面积。它们成功地用在几种熟知的芯片中，且是 DRAM 和其他重要计算机元件的基础。可惜的是它们的功耗可能很大，这也许会限制它们的应用。

在动态电路中，时钟 ϕ 定义在每个周期中的预充电和求值操作。因为电荷不能保持在电容节点上，每个预充电周期将从电压源中"拉"出电流，从而增加了电路的总功耗。时钟电路本身要求动态功率来驱动 FET。在标准结构中，每级相应于预充电管和求值管对时钟驱动器呈现如下电容：

$$C_L = C_{Gp} + C_{Gn} \tag{9.34}$$

单时钟电路的功耗就可能是总功耗的很大一部分。

VLSI 系统设计常常由于芯片的总功耗而变得很复杂。这影响到封装的选择，所希望的应用(桌上系统或便携系统)，电源特性以及吸热和机箱通风的要求。在系统约束和电路设计之间的权衡应当总是设计的考虑因素。

9.6 双轨逻辑电路

我们已经集中讨论了单轨逻辑电路，即电路中一个变量的值或者只是 0，或者是 1。在双轨电路中，两个变量即 x 和它的反信号 $\bar{x}$ 用来形成差：

$$f_x = (x - \bar{x}) \tag{9.35}$$

利用这个差 f_x 可提高开关速度。这可通过计算以下对时间的导数

$$\frac{df_x}{dt} = \left(\frac{dx}{dt} - \frac{d\bar{x}}{dt}\right) \tag{9.36}$$

并且注意到

$$\frac{d\bar{x}}{dt} \approx -\left|\frac{dx}{dt}\right| \tag{9.37}$$

来理解，因为 x 增加时 $\bar{x}$ 减少，反之亦然。于是

$$\frac{df_x}{dt} \approx 2\left|\frac{dx}{dt}\right| \tag{9.38}$$

所以 f_x 的变化率大约是单个变量的两倍。用逻辑术语来说，这意味着开关速度几乎是单轨电路所能达到的两倍那么快。

双轨电路复杂的原因是电路复杂性的提高及布线开销。每个输入和输出的数目加倍，包含变量本身及它的反信号。电路相应地比较复杂，它的设计也要有技巧。然而它们速度的优势便值得去研究它们。有些研究甚至提供了结构化的和紧凑的版图设计方法。

9.6.1 CVSL

大多数双轨 CMOS 电路多半基于"**差分串联电压开关逻辑**"，后者是 **DCVS 逻辑**或**差分 CVSL** 的同名词；这里将采用后一个名词。CVSL 是双轨逻辑门，它们的电路本身包含锁存特性。输出结果 f 和 $\bar{f}$ 一直维持到输入发生变化时为止。

CVSL 逻辑门的基本结构显示在图 9.27 中。输入包括变量（a,b,c）和它们的反信号（$\bar{a},\bar{b},\bar{c}$），它们被连接到 nFET"逻辑树"电路中。逻辑树可以模拟一对互补的开关 Sw1 和 Sw2，它们由输入决定其中一个闭合而另一个断开。开关的状态确定了输出，例如，Sw1 闭合则 $f=0$，另外一边（$\bar{f}$）由于 pFET 锁存器的作用而变为反相状态（$\bar{f}=1$）。

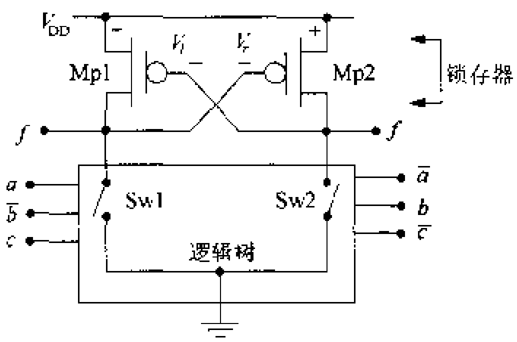


图 9.27 CVSL 逻辑门的结构

这个锁存器是由左边和右边的源-栅电压 V_l 和 V_r 控制的。假设 Sw2 闭合，使右边的 $\bar{f}=0$。此时，

$$V_l = V_{DD} \tag{9.39}$$

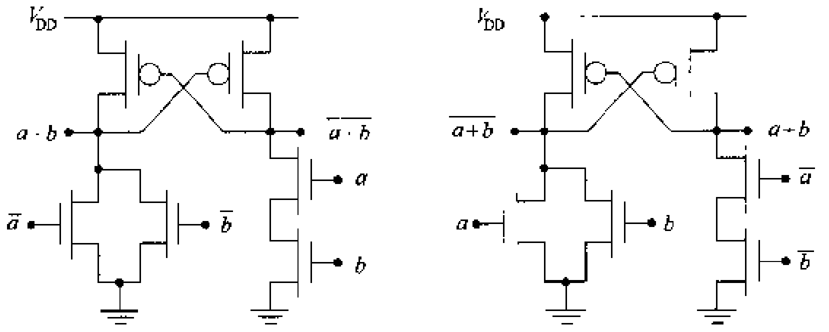
它使 Mp1 导通。当 Mp1 导通时，左边的输出节点有一条到电源的路径，使该节点为 V_{DD}；这就是 $f=1$ 状态。利用一边的下拉能使锁存器置位，有助于这个电路的迅速响应。

已经发表几种设计这个逻辑电路的技术。一种简单的方法是左边和右边采用各自的电路。图 9.28（a）是一个 AND / NAND 电路，其右边的输入为（a,b）左边的输入为（$\bar{a},\bar{b}$）；记住双轨逻辑门要求成对互为反相的输入和输出。右边 NAND 操作采用串联的 nFET 实现，这与在标准 CMOS 中的 nFET 逻辑完全一样。为了得到左边的电路，只需要用 DeMogan 恒等式：

$$\overline{a \cdot b} = \bar{a} + \bar{b} \tag{9.40}$$

由圆圈推移法的学习知道，上式表示两个 nFET 并联，输入为反相信号。图 9.28（b）是一个 OR / NOR 电路。逻辑的实现也与 AND / NAND 电路的方法相同。一个较为重要的观察是 OR / NOR 和 AND /NAND 门在形式上是相同的，只是输入的位置不同。这个对称性是由

于 OR 和 AND 是逻辑对偶的缘故。



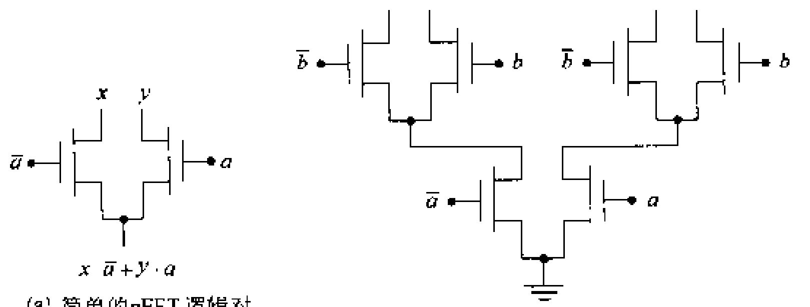
(a) AND/NAND (b) OR/NOR

图 9.28 CVSL 门的例子

逻辑树提供一种较为结构化的方法来设计开关电路。这些电路基于互为相反的输入驱动的 nFET 对，如图 9.29 (a)所示。x 和 y 应用到一对管子的顶部，这对管子的作用类似于一个 2:1 的 MUX，其(底部)的输出为：

$$x \cdot \overline{a} + y \cdot a \tag{9.41}$$

从定性上说，如果 $a=0$，则传送 x；而 $a=1$，则输出为 y。因此这对信号($\overline{a}$，a)相当于输入图案为(0,1)，这与在功能表中列出输入组合的方式是相同的。如果 $x=y$，那么输出总是 x，因此 FET 可以去除。图 9.29 (b)为一个两层的 nFET 对，上下叠在一起。上面一行的一对 b 输入相应于输入序列(0 1)(0 1)，而底下一对(a 输入)的输入序列为(0 1)。这就形成了从一个两输入的逻辑功能表到 nFET 阵列的一一映射关系。



(a) 简单的nFET逻辑对 (b) 堆叠的逻辑对

图 9.29 nFET 逻辑对

图 9.30 的门是一个例子。真值表中输出 f 的序列(1001)表明对于 $f=1$ 为 XOR 功能，而对于 $f=0$ 是 XNOR 功能。将这个表格映射就得到所示的树结构。把 pFET 锁存电路加入 f 和 $\overline{f}$ 线上就完成了 CVSL 门。这个方法可以应用到具有几个变量的任意功能表中。多余的信号对可以去除，以构成紧凑的电路。

图 9.31 为一个动态 CVSL 电路。它用钟控 pFET 代替静态锁存器，这对钟控 pFET 用来预充电输出节点。树结构底部的一个 nFET 用于求值。在电路图中采用一种简化的表示方法。每个“－ ＋”方块相应于一对 nFET，输入变量加在“＋”一边，而它的反信号加在“－”一边。在把功能表转化为逻辑树时采用了两次简化，因为 f 左半部分的序列为 00 11，因此可以取消 c 层次上的两对管子。

f	1	0	0	1
b	0	1	0	1
a	0	0	1	1

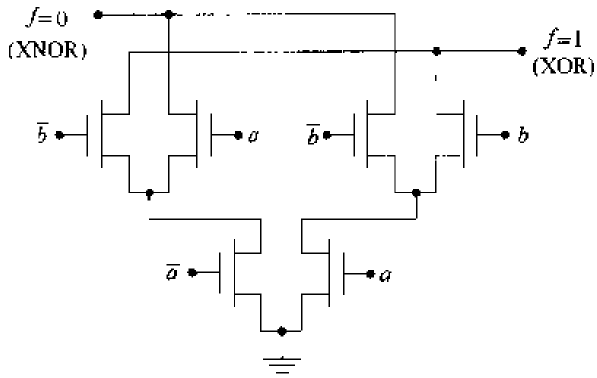
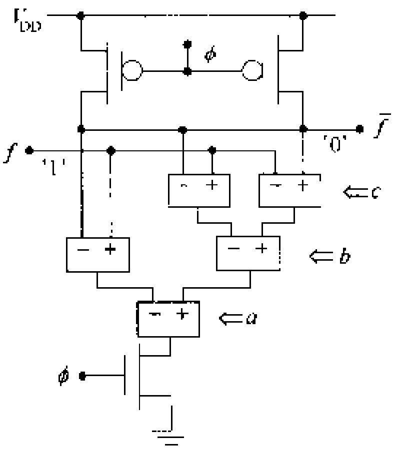


图 9.30　用 nFET 对构成逻辑树的例子



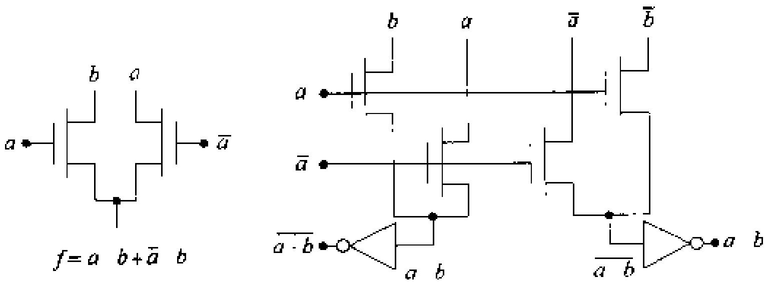
f	0	0	1	1	0	1	1	0
c	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
a	0	0	0	0	1	1	1	1

图 9.31　具有 3 层逻辑树的动态 CVSL 电路

9.6.2　互补传输管逻辑

互补传输管逻辑(**CPL**)是基于 nFET 逻辑方程的一种有趣的双轨技术。让我们考察一下图 9. 32 (a)中的 nFET 对,输出为:

$$f = a\cdot b+\bar{a}\cdot a \tag{9.42}$$



(a) AND门　　(b) AND/NAND阵列

图 9.32　CPL AND / NAND 电路

逻辑上,上式可简化为 AND 操作 $f=a\cdot b$,因为 $\bar{a}\cdot a = 0$。加入右边的晶体管是为了保证当 $a=0$ 时输出 $f=0$ 是一个严格定义的电路电压(来自输入 a)。这是传输管逻辑的基础。

为了构成 CPL,必须加入 NAND 操作。这是用 AND / NAND 对来实现的,如图 9.32 (b)所示。通过如下简化可以得到 NAND 操作:

$$a\cdot\bar{b}+\bar{a}=\bar{a}+\bar{b}=\overline{a\cdot b} \tag{9.43}$$

由于 nFET 有阈值损失,所以加入静态输出反相器把电压恢复到全幅值。在要求有个全幅的电源电压值之前,这些实际上不一定有必要,但它们也有助于加速电路。

CPL 的一个突出特点是采用相同的晶体管拓扑连接但具有不同的输入顺序,可以构成几种两输入的门。图 9.33 (a)显示一个 OR / NOR 阵列。将它与 AND / NAND 比较,表明只是把 FET 输入端的 a 和 $\bar{a}$ 互换。图 9.33 (b)为一对 XOR / XNOR。它的实现只是通过改变顶部(漏端)的输入。CPL 也可以构成具有类似特性的 3 输入逻辑门。

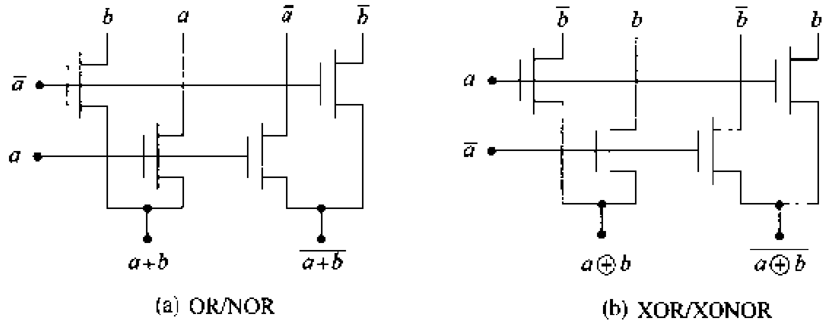


图 9.33 2输入 CPL 阵列

CPL 是一种有趣的方法,因为它提供简洁的逻辑门,并且单元版图可以复用;主要的缺点是阈值损失以及输入变量也许需要驱动一个以上 FET 的端口。在文献中已经提议为克服这些问题而设计的类似方法,但所有这些都导致比较复杂的电路。

9.7 参考资料

[1] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Press, Norwell, MA, 1995.

[2] Kerry Bernstein, et al, **High Speed CMOS Design Styles**, Kluwer Academic Press, Norwell, MA, 1998.

[3] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2001.

[4] Jan Rabaey, **Digital Integrated Circuits**, Prentice Hall, Upper Saddle River, NJ, 1996.

[5] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Press, Norwell, MA, 1999.

[6] Neil H. E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, Reading, MA, 1993.

9.8 习题

[9.1] 你的一个同学决定采用镜像电路来实现图 P9.1 真值表所描述的两个输入逻辑

函数。

(a)　这一函数是否具有构成一个镜像电路所需要的正确的对称性？如果具有，构成这个逻辑门。

(b)　对这一情形，镜像电路是否是一个聪明的设计？请解释。

[9.2]　两个串联的 pFET 有一个公共的电容 48 fF，如图 P9.2 所示。管子的 $\beta_p = 250$ $\mu A/V^2$ 及 $(V_{DD} - |V_{Tp}|) = 2.65$ V。用这两个管子同时设计一个标准的 AOI XOR 电路及一个镜像类型的 XOR 电路，设在输出节点处总的输出电容为 $C_{out} = 175$ fF，求这两个设计的 t_{LH} 值。

a	b	f
0	0	0
0	1	1
1	0	0
1	1	1

图 P9.1

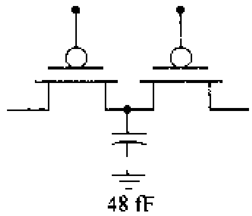


图 P9.2

[9.3]　考虑一个 CMOS 工艺，它的特征参数是 $V_{DD} = 5$ V，$V_{Tn} = 0.7$ V，$V_{Tp} = -0.85$ V，$k'_n = 120$ $\mu A/V^2$ 及 $k'_p = 55$ $\mu A/V^2$。采用宽长比为 4 的 nFET，设计一个准 nMOS 反相器。

(a) 求使 $V_{OL} = 0.3$ V 所需要的 pFET 宽长比。

(b) 假设选择 pFET 的宽长比是 $(W/L)_p = 3$，求这一情形下的 V_{OL}。

[9.4]　对于 9.3 题所描述的工艺，设计一个 NAND2 门和一个 NAND3 门，两个门都有 $V_{OL} = 0.4$ V，pFET 的宽长比为 2。然后比较这两个门的管子面积。

[9.5]　画出能实现如下逻辑操作的准 nMOS 电路：

(a) $f = \overline{a \cdot b + c}$；(b) $h = \overline{(a + b + c) \cdot x + y \cdot z}$；(c) $F = \overline{a + (c \cdot [x + (y \cdot z)])}$

[9.6]　考虑如下两个对偶表达式：

$$g = \overline{x \cdot y + z \cdot w} \qquad G = \overline{(x + y) \cdot (z + w)} \tag{9.44}$$

当采用准 nMOS 设计时哪种形式(AOI 或 OAI)可以提供最佳性能？

[9.7]　设计一个三态电路，它在控制信号 $T = 1$ 时处于高阻状态，而在 $T = 0$ 时其作用如一个非反相的缓冲器。

[9.8]　设计一个钟控 CMOS 电路实现如下功能：

$$f = \overline{a \cdot (b + c) + x \cdot y} \tag{9.45}$$

[9.9]　一个 C^2MOS 电路的输出节点在时钟信号 $\phi = 0$ 时处于三态。在该节点上的输出电容为 $C_{out} = 76$ fF。漏电电流估计为 $i_n = 0.46$ μA 和 $i_p = 127$ nA。输出电压必须保持在 2.4 V 以上，以被下一级确认为逻辑 1。

(a) 如果 $V_{DD} = 5$ V，求输出节点的维持时间。

(b) 如果 $V_{DD} = 3.3$ V，求输出节点的维持时间。

[9.10]　考虑一个如下形式的电荷泄漏方程：

$$I_L(V) = -C_{out}\frac{dV}{dt} \tag{9.46}$$

式中 C_{out}为常数,而泄漏电流由下式描述

$$I_L(V) = B\frac{V}{V_0} \tag{9.47}$$

式中 B 和 V_0 为常数。

(a) 运用 $V(0)=V_0$,解这个微分方程求出 $V(t)$。

(b) 如果最小的逻辑 1 电压是 $V_x=0.4\ V_0$,求维持时间 t_h 的表达式。

[**9.11**] 画出具有如下输出的一个动态逻辑门的电路图,采用最少数目的管子:

$$f = \overline{a \cdot b + c \cdot a} \tag{9.48}$$

[**9.12**] 画出具有如下输出的一个动态逻辑门的电路图:

$$F = \overline{a \cdot (b+c+d)} \tag{9.49}$$

[**9.13**] 图 P9. 3 中存放在 100 fF 电容上的输出电压在 $A=B=0$ 时的初始值为 5 V。如果信号改变为 $A=0, B=1$,求 V_{out}的值。

[**9.14**] 4 个 nFET 作为传输管,如图 P9.4 所示。输入电压设为 $V_{in}=V_{DD}=5$ V。若 $V_{Tn}=0.75$ V

(a) 对第 1 种情形,假设信号最初为$(A,B,C,D)=(1,1,0,0)$,然后转变为$(A,B,C,D)=(0,1,1,1)$,求 V_{out}的最终值。

(b) 假设信号最初为$(A,B,C,D)=(1,1,1,0)$,然后转变为$(A,B,C,D)=(0,0,1,1)$求 V_{out}的最终值。

[**9.15**] 设计一个 MODL 电路,实现如下两个输出:

$$F = a \cdot b \qquad G = (a \cdot b)\cdot(c+d) \tag{9.50}$$

[**9.16**] 通过建立 nFET 逻辑树结构,就图 P9.5 所示功能表设计一个 CVSL 门。

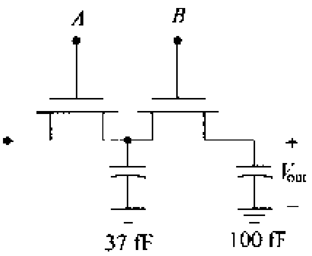


图 P9.3

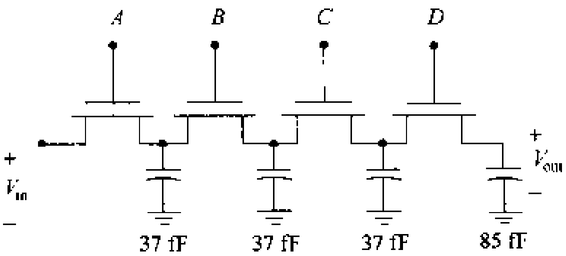


图 P9.4

f	1	1	0	1	0	0	1	1
c	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
a	0	0	0	0	1	1	1	1

图 P9.5

第 3 部分



VLSI 系统设计

第10章 用Verilog® 硬件描述语言描述系统

硬件描述语言(HDL)是一个层次化设计的理想工具。一个系统的描述可从最高抽象的体系结构层次向下到最基本的逻辑门和开关。

这个领域有两种最主要的硬件描述语言：VHDL(VHSIC HDL)① 和 Verilog® HDL。VHDL起始于政府力图把各个不同的合同项目统一起来，而Verilog则是私营企业开发的成果。这两者都已标准化并且广泛地用于工业界，所以本书可以介绍任何一种。之所以选择Verilog，是因为它在VLSI设计中非常普遍。与VHDL相比，它是一种相对较不严谨和编程自由的语言，而且大多数芯片设计者感到它更贴近他们思考的方式。Verilog是在C语言之后形成的，并且运用类似的过程和结构。然而注意，C或C++它们本身也是可以用来作为HDL的[9]，而且有几家公司也开发它们自己的语言。

本章介绍Verilog语言的基本概念。如果从其他课程已经熟悉VHDL，那么你将发现学习Verilog是很容易的。如果只是首次涉足HDL，你不必担心；路程是平坦的，这些概念很容易掌握。

10.1 基本概念

硬件描述语言可以采用词和符号而不必采用图形表示法，如功能块或逻辑图来描述构成数字系统的部件。每个部件是由它的输入和输出端，所实现的逻辑功能是以时序特性如延时和时钟来定义的。整个数字系统可以用本文形式采用预先规定的一组规则和关键词(保留词)来描述。这一文件经语言编辑器处理，其输出可以进行分析以得到正确的工作。这可以应用到简单逻辑门或应用到整个微处理器设计中。通常认为运用HDL进行逻辑验证是必须进行的，以确认设计是否正确。

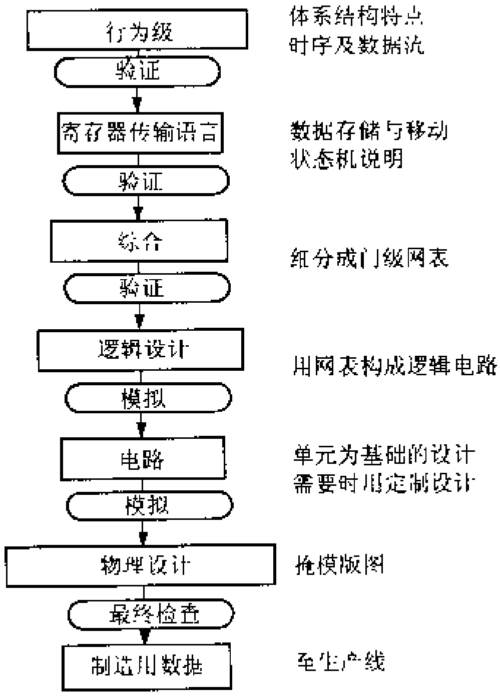


图10.1 VLSI设计流程的例子

图10.1为一个典型的层次化设计过程。在最高层次上是行为描述，它描述一个系统的体系结构特点。这一般来说是非常抽象的，它并不包含有关如何具体实现这一设计的任何细节。一旦行为模型被模拟和优化，设计就向下进入**寄存器传输级(RTL)**。一个数字电路的RTL描述集中在数据是

---

① VHSIC是DoD的非常高速集成电路的缩写；而DoD是美国国防部的缩写。

如何在系统中从单元到单元移动的,以及所进行的主要操作。在这一层次上可以引入状态机和时序电路。时序的上下限窗口被检查和再检查,同时系统的验证仍然是一个最基本的任务。

设计过程的下一个层次称为**综合**。在全自动设计中,RTL 描述被送入一个综合工具,它产生实际构造这个系统所需硬件部件的网表。最为普遍使用的综合工具之一是 Synopsis®。综合过程成功或失败与否常取决于 RTL 编码的技巧。不是所有的 HDL 结构都能被综合,一个典型的估计大约在 50%附近。

在综合步骤之后,网表被用来设计逻辑电路。在这一层次上的验证包括进行模拟,以保证逻辑是正确的。一旦逻辑被验证,就可以用单元库来设计电路。部件用导线连接在一起,并且通过模拟验证它们的电气特性和逻辑功能。在物理设计阶段,单元例图和布线被转换成硅图案。在版图验证之后,这一设计(最终!)完成并且送去制造第一块硅试验芯片。

Verilog HDL 提供数字系统在上面列出的所有层次上的描述。每个层次都是与每个其他层次相关联的,而层次化设计的原理是通过各种不同类型的代码联系在一起的。每个层次运用确定的一组命令和结构,并且有它自己的编码形式。Verilog 甚至提供 MOSFET 的开关级模型,虽然它不如电路模拟器,如 SPICE 的描述那么具体,对 CMOS 工艺变量那么敏感。Verilog-A 是一个原本数字系统的描述语言向模拟系统的延伸。

联系不同层次的概念是一个**模块**的概念。一个 Verilog 模块对执行某种功能的单元进行描述。它可以如基本的 FET 开关那么简单,也可以像一个 64 位的 ALU 那么复杂。可用简单模块的例举来构建更为复杂的模块。这里的层次化结构非常类似于本书前面所描述的在版图编辑器中设计单元时所用的层次化结构。

对 Verilog 的描述从数字逻辑层次开始。在这个层次上,用简单的门来构成较为复杂的逻辑单元。一旦理解这个语言的结构之后,将介绍更高的抽象层次。

10.2 结构化的门级模型

结构化的模型以构成系统的部件,描述一个数字逻辑电路。门级模型基于运用最基本的逻辑门,说明它们是如何连在一起的。它最容易学习,因为它与在基本逻辑中介绍的概念是类似的。

Verilog 是用编辑器理解的一定的关键词构成的。在这组关键词中有基本元件(如逻辑门),信号类型以及命令。在列出的 Verilog 代码中采用如“this type”这样的字体(称为 sans serif 字体),而且将从正文缩进。关键词将用同样字体的**黑体字**。在结构模型层次上,关键词常常是最基本的逻辑操作(门),这会形成非常易读的编码风格。学习 Verilog 的一个直接方法是通过一行一行的代码分析,学习如何将一个逻辑电路转换成 Verilog 描述。我们将以直接的方式来说明这些概念和句法。

Verilog 举例

考虑一个 4 输入的 AOI 电路,如图 10.2 所示。这个门是用基本的 AND 和 NOR 门构成的,输入为 a,b,c 和 d,产生的输出为

$$f = \mathrm{NOT}(a \cdot b + c \cdot d) \tag{10.1}$$

让我们来考察 Verilog 模块代码是如何用它内部的结构来描述这个电路的,然后通过细节来学习这个模块是如何构成的。

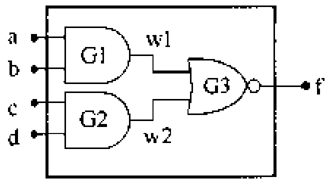


图 10.2　AOI 模块例子

```
module AOI4(f,a,b,c,d);
  input a,b,c,d;
  output f;
  wire w1,w2;
  and G1(w1,a,b);
  and G2(w2,c,d);
  nor G3(f,w1,w2);
endmodule
```

我们来初看一下所列出的代码,它说明了一个 Verilog 模块的结构和句法。关键词 **module** 定义一个名为 AOI4 电路代码的开始。代码的最后一行 **endmodule** 则表明这个模块的描述已经完成。输出和输入"标识符"的名字列在括号中,其中输出 f 第一个列出,然后是输入 a,b,c,d。在 Verilog 中用分号作为分隔符。它们的用途应该记住。

接下来的几行代码是端口关键词 **input** 和 **output**,它们说明输入和输出变量。关键词 **wire** 说明 w1,w2 是描述这个电路所需要的内部值,但它们不是 **input** 和 **output** 端口。**wire** 说明一种称为"节点(net)"的数据类型。一个节点的值是由驱动门的输出来决定的。在这里,w1 和 w2 是 AND2 门的输出,而它们又由输入值来决定。

逻辑结构由接下来的三行说明。它们是 AND 和 NOR(原文为 NOT——译者注)基本门的例举,这些门是 Verilog 语言的一部分。一个门的例举具有以下形式:

```
gate_name instance_name(out,in_1,in_2,in_3,...);
```

式中 instance_name 是一个可选的说明符,它把联系逻辑门的类型和所列出的逻辑门的名字联系起来。在例子中,已经命名门 G1,G2 和 G3,所以它们出现在列出的代码中。如果把它们略去不写,那么编译器会以同样的方式解释代码。

这一结构描述提供与逻辑电路部件惟一的一一对应关系。假设从下面的模块描述开始,然后用它构建逻辑图。

```
module Example(s_out,c_out,in_0,in_1);
  input in_0,in_1;
  output s_out,c_out;
  xor(s_out,in_0,in_1);
  and(c_out,in_0,in_1);
endmodule
```

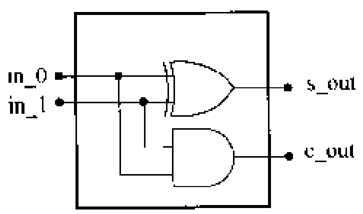


图 10.3　从列出的 Verilog 代码构成逻辑电路

这将生成图 10.3 所示的内部细节。这从输入端口 in_0 和 in_1 开始画,加上门(xor 和 and)以及所说明的连线,然后从这一模块的中心部分引出输出(s_out 和 c_out)。逻辑方程为:

$$s_out = (in_0) \oplus (in_1)$$
$$c_out = (in_0 \cdot in_1)$$

可以看出,这是一个半加器的"和"与"进位输出"。这些例子说明 Verilog 的结构描述等同于在一个标准逻辑方块图中所含的信息。

在做进一步介绍之前,让我们考察一下写 Verilog 描述的一些基础。

1. 标识符(Identifiers)

标识符是模块、变量以及在设计中可以参照的其他对象的名字。至此已用的标识符例子包括 AOI4,a,b,in _ 0 和 s _ out。标识符可以含有大写和小写字母,数字 0 至 9,下划符号(_)和美元符号($)。在通常运用中,第 1 个符号必须是一个字母或下划符。一个标识符必须是连在一起的一组符号。例如 input _ control _ A 是一个标识符,而 input control A 则不能作为一个标识符。

应当指出 Verilog 语言**对大小写敏感**。必须仔细,不要混淆大小写字母,因为它们是指不同的东西。例如 in _ 0,In _ 0,和 IN _ 0 是完全不同的,因而不能互换。列出代码时对于空格是不敏感的,可以插入许多空格或空行以提高可读性。

2. 数组 (Value Set)

数组是指一个二进制变量可以具有的具体值。Verilog 提供为描述硬件所需要的 4 种值:0,1,x 和 z。0 和 1 电平是通常的二进制值。一个 0 或者表示一个逻辑 0 或者表明一条语句不成立,而 1 或者表示一个逻辑 1 或者表明一条语句成立,由上下文来确定究竟做何解释。x 代表一个未知值,而 z 是高阻(Hi - Z)值。未知值 x 是很重要的,因为在许多情形中并没有足够的信息。例如一开始使一个电路上电时,逻辑门的输出是未知的;必须等待一组输入建立起它们的值。

除了这 4 种值而外,0 和 1 值还可以再分为 8 个"强度"。它们用来模拟各种各样的物理现象,这些现象会使参与竞争的一条控制线的信号强度降低。强度将在后面进行细节讨论。

3. 基本门(Gate Primitives)

基本逻辑功能的关键词,提供在这一层次上建立结构模型的基础。在 Verilog 中重要的操作是 **and,nand,or,nor,xor,xnor,not** 以及 **buf**,这里 **buf** 是一个非反相的驱动缓冲器。除了 not 和 buf 以外,所有的门都可以具有 2 个或以上的输入。

对 0 和 1 输入的真值表是以通常方式定义的。然而由于允许有 x 和 z 的值,必须定义一个门对于扩充的一组输入激励是如何响应的。buf 和 not 门如图 10.4 所示的表格来定义。在表格上面一行中的输入值产生在第 2 行中的输出值,这些是不言自明的。

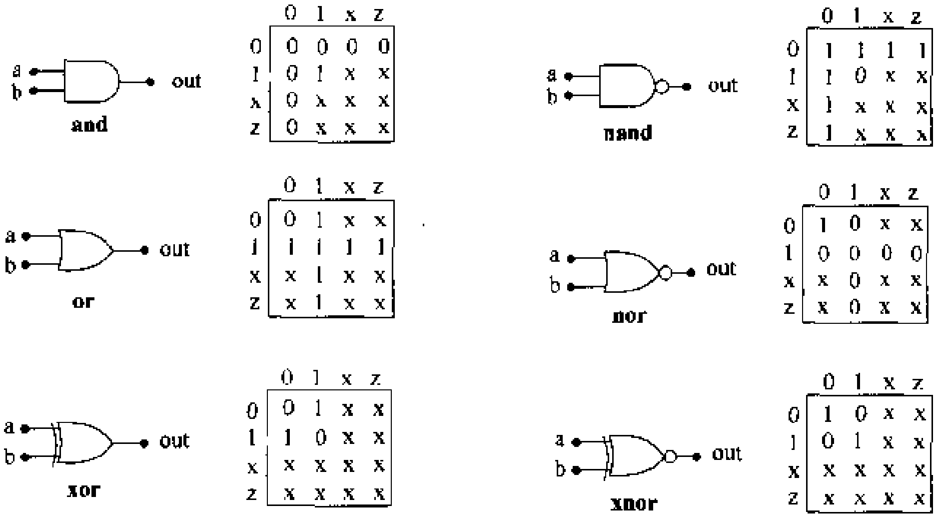


(a) **buf**基本单元　　(b) **not**基本单元

图 10.4 **buf** 和 **not** 门的功能对应图

图 10.5 提供了多输入门 **and,nand,or,nor,xor** 和 **xnor** 的真值表。这些表格本身针对两个输入,但可以推广适合于 3 个或更多的输入。表格的形式是 Verilog 的标准形式,并且具有卡诺图的结构。最上面一行是其中一个输入的值,而左边一列则是另一个输入的值。每种可能的输出值 out 是从包含在方框中的矩阵通过对准一行和一列来读出的。很容易看出左上角

的 2×2(原文为 4×4——译者注)子矩阵是一个针对输入为 0 和 1 的标准卡诺图。



and:

	0	1	x	z
0	0	0	0	0
1	0	1	x	x
x	0	x	x	x
z	0	x	x	x

nand:

	0	1	x	z
0	1	1	1	1
1	1	0	x	x
x	1	x	x	x
z	1	x	x	x

or:

	0	1	x	z
0	0	1	x	x
1	1	1	1	1
x	x	1	x	x
z	x	1	x	x

nor:

	0	1	x	z
0	1	0	x	x
1	0	0	0	0
x	x	0	x	x
z	x	0	x	x

xor:

	0	1	x	z
0	0	1	x	x
1	1	0	x	x
x	x	x	x	x
z	x	x	x	x

xnor:

	0	1	x	z
0	1	0	x	x
1	0	1	x	x
x	x	x	x	x
z	x	x	x	x

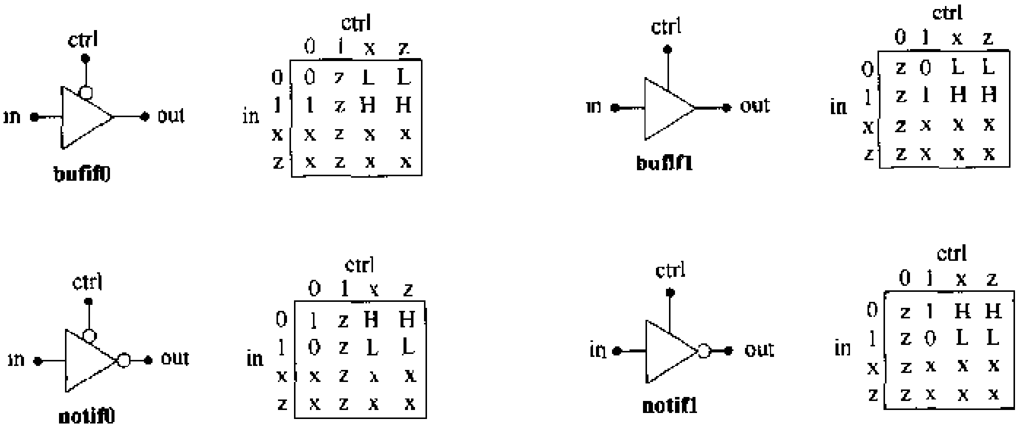
图 10.5　多输入门的对应图

三态的基本门是 **bufif0**, **bufif1**, **notif0** 和 **notif1**。它们的名字有利于记住它们的操作。bufif0在控制为 0 时为一个缓冲器;如果控制为 1,那么它就处于三态,其输出为 Hi - Z(高阻态)。类似地,**notif1** 在控制为 1 时其作用如同 **not**,而控制为 0 时输出为高阻态。三态门有一个输入,但它们用做驱动器时可以有一个以上的输出。我们用以下形式来描述三态门:

```
tristate_name instance_name(out_0,out_1,out_2,...,input,control);
```

式中 instance_name 是可选择的例图名字。图 10.6 概括了这些基本门的逻辑映射。图 10.7 是一个 2:1 MUX 的三态电路的例子。这个电路的逻辑表达式为:

$$out = p_0 \cdot \bar{s} + p_1 \cdot s \qquad (10.2)$$



bufif0 (行: in, 列: ctrl):

in \ ctrl	0	1	x	z
0	0	z	L	L
1	1	z	H	H
x	x	z	x	x
z	x	z	x	x

bufif1:

in \ ctrl	0	1	x	z
0	z	0	L	L
1	z	1	H	H
x	z	x	x	x
z	z	x	x	x

notif0:

in \ ctrl	0	1	x	z
0	1	z	H	H
1	0	z	L	L
x	x	z	x	x
z	x	z	x	x

notif1:

in \ ctrl	0	1	x	z
0	z	1	H	H
1	z	0	L	L
x	z	x	x	x
z	z	x	x	x

图 10.6　三态基本门的对应图

它可以用下面列出的 Verilog 代码来描述:

```
module 2_1_mux(out,p0,p1,s);
  input p0,p1,s;
  output out;
  bufif0(mux_out,p0,s);
```

```
    bufif1(mux_out,p1,s);
endmodule
```

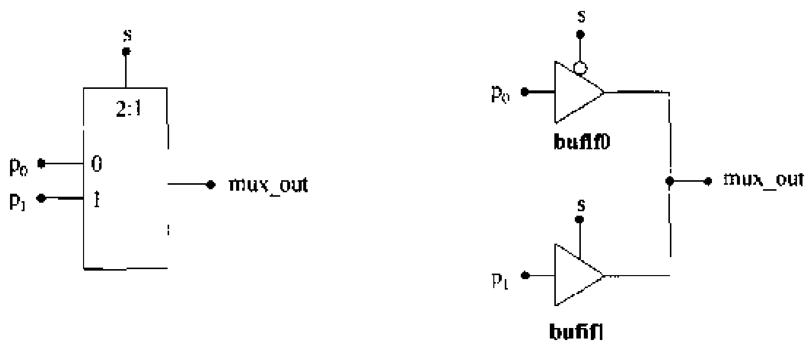


图 10.7 采用三态基本门构成 2:1 MUX

其他基本单元将在后面介绍。它们包括 MOSFET 开关及其他有用的部件。

4. 注释行(Comment Lines)

注释行对于代码归档非常有用。在下述语句中：

```
xor(s_out,in_0,in_1);//This line produces s_out
```

在//符号右边的每个词均被编译器忽略。如果注释延伸到两行或更多的行,那么在延伸的第 1 行中用/* 来表示注释的开始,而在最后一行中用 */ 表示结束,如：

```
/* If we have a long comment that we want to insert
   then we may extend it into multiple lines
   or whatever is convenient */
```

第 2 行中的缩进用来提高可读性,但这是可选的。注释不能嵌套在其他注释中。

5. 端口(Ports)

端口是一个模块与其他模块进行通信的接口。这相应于在库单元中的输入和输出点,所有的端口必须在列出的模块代码中说明。至此,所举的例子一直具有如下形式：

```
input in_0,in_1;
output s_out,c_out;
```

一个双向端口用以下的句法来说明

```
inout IO_0,IO_1;
```

式中标识符 IO_0 和 IO_1 既可用做模块的输入也可以用做它的输出。

接下来考虑图 10.8 的基于 NOR 门的 SR 锁存器。这一电路的 Verilog 模块描述可以写成如下的形式：

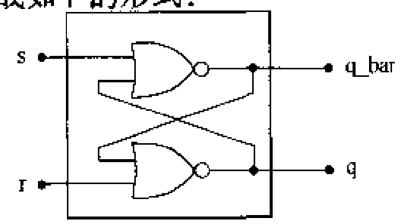


图 10.8 SR 锁存器

```
  module sr_latch(q,q_bar,s,r);
    input s,r;
  output q,q_bar;
  reg q,q_bar;
  nor(q_bar,s,q),(q,r,q_bar);
endmodule
```

这里有两个新的特点。第 1 个是寄存器(**reg**)数据类型的说明。对于一个寄存器数据类型,它的值一直保持到它被另一个值重写为止。在这里,它使 q 和 q _ bar 的值保持,以便与另一个不同模块的一个端口进行通信。注意在代码中 q 和 q _ bar 被同时说明成 **reg** 和 **output** 端口。一个 Verilog 的 **reg** 数据类型不应当被解释成一个硬件寄存器,如 D 类型的触发器,而只是把它们想像成在没有任何外部驱动器时仍能保持其值的线,一个 **reg** 属于节点类型的说明。

第 2 个新的特点是用一行来说明 **nor** 基本门的多个例举。因为这两个 nor 门的输出和输入表是不同的,每个表都作为一组列在一对括号之内,且用逗号作为它们之间的分隔符。因此单用一行就代表了两个独立的门。这一方法可以扩展到多个门,每个门都包括一个例图的名字以利于解释代码。

6. 门延迟(Gate Delay)

一个硬件描述语言采用的模型必须能在模拟时包括时延。Verilog 提供了几种方法在门一级引入延时。

经过一个门的逻辑延时常常采用从输入到输出的单个延迟时间(传输延时)来模拟。延迟在例举中是采用#号说明的,如

```
nand #(prop_delay)G1(output,in_a,in_b);
```

这里 prop _ delay 是延时值。如果分别已知上升和下降时间,它们也可以写成

```
nand #(t_rise,t_fall)G1(output,in_a,in_b);
```

关断延时也可包括在内,如

```
nand #(t_rise,t_fall,t_off)G1(output,in_a,in_b)
```

在 # 后面列出值的个数确定了 Verilog 说明这些信息的方式。只有一项表示是传播延时,两项表示 t_r 和 t_f 值,而三项则再加入关断时间。

门延时的数值用内部时间步的整数倍来说明,例如

```
and #(4,2)A1(out,A_in,B_in);
```

则表示 t _ rise = 4 个单位和 t _ fall = 2 个单位。采用相对单位足以适合很多种类的模拟,,所以没有必要用绝对时间值(秒)。

如果希望确切的时间值,那么可以列出如下形式的编译器指令:

```
'timescale t_unit/t_precision
```

在这一表达式中,t _ unit 和 t _ precision 可以具有 1,10 或 100 的值,之后便是时间尺度单位 s,ms,μs,ns,ps,或 fs 分别表示秒,毫秒,微秒,纳秒,皮秒或飞秒。t _ unit 为时间尺度,而 t _ precision 为时间尺度精度;显然 t _ unit>t _ precision。例如

```
'timescale 1ns/100ps
```

表示时间尺度每单位为 1ns, 而时间尺度的精度为 100ps。如果一个门的例子写成

```
xor #(10)(out,A_0,A_1);
```

那么通过这个门的绝对延时为 10×t _ unit = 10ns。如果我们改变时间尺度为

```
'timescale 10ns/1ns
```

那么绝对延时为 10 × 10ns = 100ns。t _ precision = 1ns 值决定了精度；例如，延时为 10.748ns，那么这个值被四舍五入到 11ns。

门延时可以检查一个电路在动态环境中的响应。让我们来模拟图 10.9 所示的输入波形为 a,b,c 的模块。下面列出的 Verilog 引入一个提供信号的激励模块的概念：

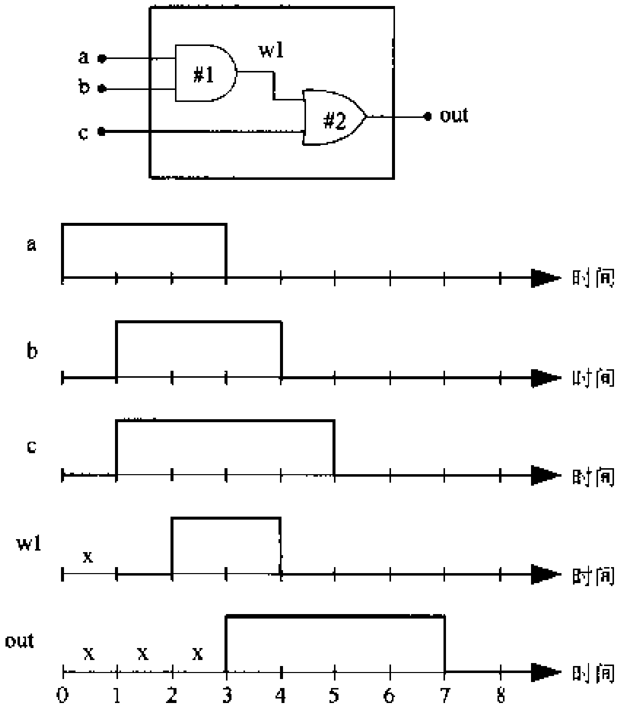


图 10.9 门延时例子

```
//这一模块包含门延时
module DelayEx(out,a,b,c);
  input a,b,c;
  output out;
  wire w1;
  and # 1(w1,a,b);
  or # 2(out,w1,c);
endmodule
//激励模块提供输入信号
module stimulus;
  reg A,B,C;//保持输入值
  wire OUT;//这是一个被驱动的输出值
//以下是例举电路
  DelayEx G1(OUT,A,B,C);
Initial
  begin
  $ monitor  ( $ time,"A= %b,B= %b,C= %b,OUT- %b",A,B,C,Out);
    A=1;B=0;C=0;
  ##1 B=1;C=1;
  #2 A=0;
  #1 B=0;
  #1 C=0;
  ##3  $ finish;
  end
```

```
endmodule
```

模块 DelayEx 的第 1 部分除了延时说明以外没有什么新的东西。激励模块可以用 Verilog 句法定义输入，"测试"模块 DelayEx。对于激励，定义变量 A，B 和 C 值为 **reg** 类型，而 OUT 是 **wire** 类型。通过以下两行使这个模块被例举到激励中去：

```
//以下是例举电路
DelayEx G1(OUT,A,B,C);
```

这里变量的次序与定义模块时一致。

下一组语句说明输入。**initial** 指令运用 **begin...end** 结构来列出时间为零时的初值。含在这一部分的是系统输出命令

```
$monitor($time,"A=%b,B=%b,C=%b,OUT=%b",A,B,C,Out);
```

式中美元符号指明这是一个编译器指令。它要求每当一个变量变化时就提供 A，B，C 和 OUT 的输出。正如后面要解释的，a = %b 表示变量 a 为二进制格式。在下一行中，输入变量的初始值对应于波形被赋值为 A=1，B=0，C=0。信号的变化用以下形式的语句**按顺序**进行描述

```
#1 B=1;C=1;
#2 A=0;
#1 B=0;
#1 C=0;
```

它们必须按次序执行。激励#1 意味着在时间 0+1=1 时，B 和 C 都为逻辑 1 值。下一行#2 则在第 1 行发生 2 个单位时间之后使 A=0，此时的绝对时间为 1+2=3 个单位。再下一行在 3+1=4 个时间单位时使 B 复位到 0，而最后一行在 4+1=5 个时间单位时使 C 复位到 0。很容易验证以上确实描写了输入波形。最后一条指令**#3 $finish** 指出模拟在 5+3=8 个时间单位时完成。最后 **end** 结束了 **begin** 结构。对此代码进行模拟，产生图中所示的输出波形。

这一例子提供了如何为 Verilog 代码建立一个测试平台的概念。一旦定义了电路，就可以写出各种激励模块来测试这个电路逻辑。这个概念表示在图 10.10 中。激励模块通常与逻辑模块分开，以便改变输入而不影响逻辑。Verilog 的工作环境使这两者在模拟时可以链接起来。具体细节会因编译器的实现而有所不同，所以阅读文档非常重要。逻辑验证是高层次 VLSI 设计最重要的特点之一。



图 10.10　测试平台概念

7. 数字说明(Number Specifications)

在前面延时的例子中,输入激励是通过如下的语句来定义的:

```
A=1;B=0;C=0;
```

在省略时这些值被解释成二进制值。这些值也可以用以下形式说明成基数为 2(二进制, b),8(八进制,o),10(十进制,d),以及 16(十六进制,h)的格式:

```
<size>'<base designator><value>
```

其中<size>是一个十进制的数,表示数中的位数。下面是一些例子:

```
1'b0       //一个 bit 的二进制数 0
4'b1011    //四个 bit 的二进制数 1011
16'h1a36   //十六个 bit 的十六进制数 1a36
3'd4       //三个 bit 的十进制数 4=100₂
```

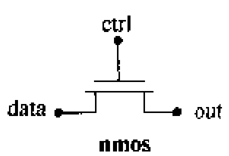
在代码中可以说明数值。例如以下的代码可按要求说明 reset 的值:

```
reg reset;
initial
    begin
    reset = 1'b1;          //初始化复位置 1 值
    # 10 reset=1'b0;       //在十个时间单位之后复位置 0
    end
```

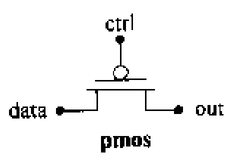
10.3 开关级建模

Verilog 允许根据 MOSFET 的行为建立开关级的模型。尽管电路级的模拟器(如 SPICE)在进行关键电气计算时要精确得多,但 Verilog 编码在验证同时包含管子和逻辑门的电路的逻辑关系时是非常有用的。更为重要的是在第 2 章中已经讨论的那样,开关级的模型与 CMOS 电路和逻辑门有直接的一一对应关系。从复杂的系统级设计一直向下到基本的 CMOS 电路,都能用 Verilog 描述,显示了层次化设计的威力。

开关级的基本单元称为 **nmos** 和 **pmos**,它们的行为方式与名字相同的管子一样。图 10.11 总结了它们的行为。这两种基本单元的 Verilog 句法具有以下形式:



data \ ctrl	0	1	x	z
0	z	0	L	L
1	z	1	H	H
x	z	x	x	x
z	z	z	z	z



data \ ctrl	0	1	x	z
0	0	z	L	L
1	1	z	H	H
x	x	z	x	x
z	z	z	z	z

图 10.11　开关级基本单元

```
nmos name(out,data,ctrl);
pmos name(out,data,ctrl);
```

式中 name 是例子的标识符,可以自己选择。对于 ctrl 值(加到栅上)为 0 和 1,其行为与 FET 一样。**nmos** 开关在 ctrl = 0 时开路,在 ctrl = 1 时闭合。**pmos** 开关在 ctrl = 0 时闭合,在 ctrl = 1 时开路。一个开路的开关产生一个高阻态,即 out = z。这些表格也列出了两个新项,L 和 H,它们分别为 ctrl = x 或 z 时 out 的值。符号 L(低)代表 0 或 z,而符号 H(高)则代表 1 或 z。这一不明确的规定并不是毫无道理的,它与输出节点可以储存电荷的概念相关,即 out 可以与它原先的值有关。

MOS 开关可以用来描述 CMOS 逻辑门。图 10.12 中简单的 NOT 电路可有如下的 Verilog 描述:

```
//CMOS 反相器开关电路
module fet_not(out,in);
  input input;
  output output;
  supply1 vdd;
  supply0 gnd;
  pmos p1(vdd,output,input);
  nmos n1(gnd,output,input);
endmodule
```

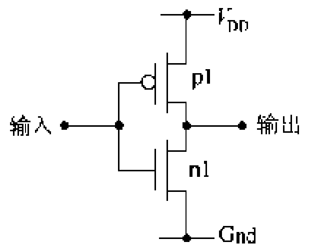


图 10.12　用 Verilog 开关构成 CMOS 反相器

这一电路和所列代码已经引入两个新的 Verilog 关键词 **supply1** 和 **supply 0**,它们定义了电源 V_{DD} 和地线 Gnd 的连接。这两个关键词分别代表最强的逻辑 1 和逻辑 0 的驱动。Verilog 模块把它们看成进入 FET 的数据,而栅的 input(输入)则是开关信号 ctrl。

同样的结构可以用来模拟随意的 CMOS 逻辑门。图 10.13 中的 NAND2 和 NOR2 开关电路是用以下模块来描述的,对 NAND 门有:

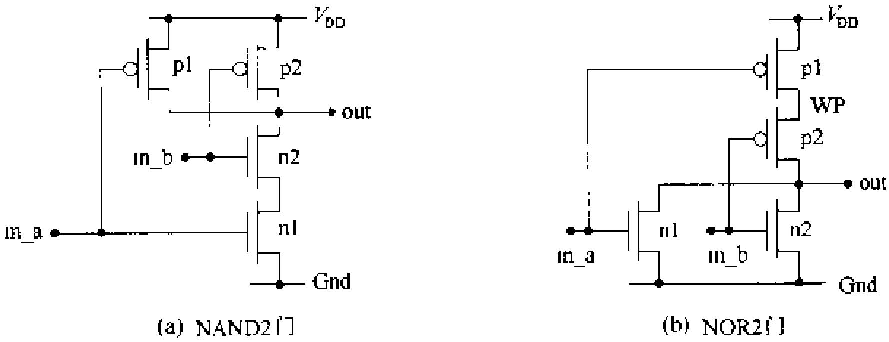


图 10.13　构成逻辑门

```
//CMOS logic gates
module fet_nand2(out,in_a,in_b);
  input in_a,in_b;
  output out;
  wire wn; //这条线连接串联的 nmos 开关
  supply1 vdd;
  supply0 gnd;
  pmos p1(vdd,out,in_a);
  pmos p2(vdd,out,in_b);
  nmos n1(gnd,wn,in_a);
```

```
  nmos n2(wn,out,in _ b);
endmodule
```

而对于 NOR 门有：

```
module fet _ nor2(out,in _ a,in _ b);
  input in _ a,in _ b;
  output out;
  wire wp; //这条线连接串联的 pmos 开关
  supply1 vdd;
  supply0 gnd;
  pmos p1(vdd,wp,in _ a);
  pmos p2(wp,out,in _ b);
  nmos n1(gnd,out,in _ a);
  nmos n2(gnd,out,in _ b);
endmodule
```

这些都可以用一行一行的比较来加以验证。

另一组有用的基本单元包括上拉和下拉部件，它们的关键词为 **Wpullup** 和 **pulldown**。它们可以模拟成连到 **supply1** 和 **supply0** 的电阻，如图 10.14(a)所示，并且可用 Verilogm 描述为：

```
pullup(out _ 1); //给出高电平输出
pulldown(out _ 0); //给出低电平输出
```

输出强度称为 **pull1** 和 **pull0**，它们要比 **supply1** 和 **supply0** 的强度弱。上拉和下拉基本单元可以用各种方式来模拟电路。例如，**pullup** 可以作为图 10.14(b)所画的 nMOS NOR3 门中的一个负载器件。其 Verilog 描述为：

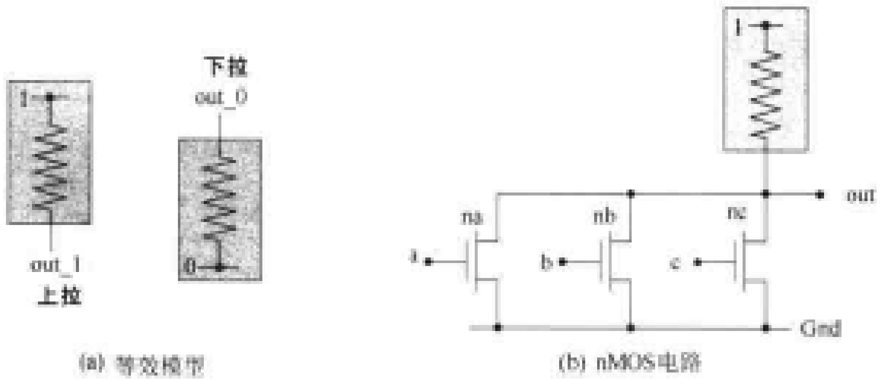


图 10.14　上拉和下拉基本单元

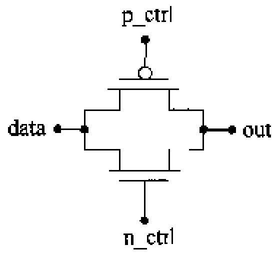
```
module fet _ nor2(out,in _ a,in _ b,in _ c);
  input in _ a,in _ b;
  output out;
  supply0 gnd;
  nmos na(gnd,out,in _ a),
       nb(gnd,out,in _ b),
       nc(gnd,out,in _ c);
  pullup(out);
endmodule
```

注意 **pullup** 和 **pulldown** 都只要求一个标识符，这是因为从每个这样的“器件”中只需提供

一条导线。

1. cmos 基本单元(The cmos Primitive)

Verilog 用关键词 **cmos** 来模拟 CMOS 传输门。② 图 10.15 表示它的符号和功能表。为了例举 TG，采用如下句法：



cmos

n_ctrl	p_ctrl	data 0	1	x	z
0	0	0	1	x	z
0	1	z	z	x	z
0	x	L	H	x	z
0	z	L	H	x	z
1	0	0	1	x	z
1	1	0	1	x	z
1	x	0	1	x	z
1	z	0	z	x	z
x	0	0	1	x	z
x	1	L	H	x	z
x	x	L	H	x	z
x	z	L	H	x	z
z	0	0	1	x	z
z	1	L	H	x	z
z	x	L	H	x	z
z	z	L	H	x	z

图 10.15　Verilog cmos 传输门

```
cmos tg1(out,data,n_ctrl,p_ctrl);
```

这里 data 是输入。在大多数情形中，n_ctrl 和 p_ctrl 是互补信号。然而，表格列出的是最一般的情况，即这两个量是独立无关的。这在实际中是可能发生的，因为当一个信号是从另一个信号产生时就会有一个反相器的延时。

2. 延迟时间(Delay Times)

时间延迟的句法与用在逻辑门中的一样。延时采用"#号(时间)"的形式以时间单位来说明。在"#号(时间)"中列出的项数确定它们的含义。如只有一项为传播延时，有两项为(t_rise,t_fall)，而有三项是指(t_rise,t_fall,t_off)。下面是一些例子。

```
nmos #(2)n1(out,data,ctrl);
pmos #(3,4)p1(out_p,data_in,p_ctrl);
cmos #(2,3,3)TG1(output,input,n_sig,p_sig);
```

这些未必与取决于负载的实际值相关，所以在说明器件延时时必须小心。

3. 强度等级(Strength Levels)

除了强度 0,1,x 和 z 而外，变量还允许取其他不同的强度。它们用在两个或更多个信号同时竞争控制一个节点的场合，或描述一个实际的电压损失。图 10.16 总结了逻辑 1 和逻辑 0 值的强度范围。当存在几个不同信号间的竞争时，则由较强的一个信号控制。强度对于模

② 注意将用小写黑体字 sans serif 字体来区分关键词 cmos 及 CMOS 工艺。

拟电压变化，如通过传输管的阈值损失是非常有用的。强度可以按要求说明，或者也可引入电阻性开关，在它们的定义中包括信号变化的特性。

逻辑1			逻辑0	
强度等级	名称	类型	名称	强度等级
supply1	Su1	drive　　（最强）	Su0	supply0
strong1	St1	drive	St0	strong0
pull1	Pu1	drive	Pu0	pull0
large1	La1	storage	La0	large0
weak1	We1	drive	We0	weak0
medium1	Me1	storage	Me0	medium0
samll1	sm1	storage	sm0	small0
high-z1	Hiz1	high-Z　　（最弱）	HiZ0	high-z0

图 10.16　Verilog 中的强度等级

4. 电阻性开关(Resistive(rmos)Switches)

现实的 MOSFET 具有漏－源电阻，它们会改变通过它们的信号强度。这些影响中的一些，可以运用电阻性 MOS 开关来考虑。电阻性开关与通常 MOS 开关的工作方式相同，均由栅控制，但这些器件会改变它们的输出强度。它们与 FET 相当的基本单元是 **rnmos**，**rpmos** 和 **rcmos**。例举的句法也与非电阻性(理想)开关相同。例如

```
rnmos #(1,2,2)fet_1(output,input,gate_ctrl);
```

它说明一个电阻性的 nFET，主要的差别是输入－输出强度的关系是由图 10.17 表格来定义的。这对于考虑实际的影响，如通过 nFET 传输管的阈值电压损失时是非常有用的。虽然在电子学层次上的 SPICE 模拟精确得多，但电阻性开关对模拟非关键路径中的开关行为是非常有用的。

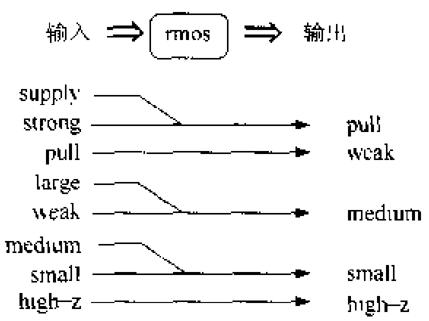


图 10.17　电阻性 MOS 管(rmos)输入-输出强度的对应关系

10.4 层次化设计

基本单元、模块和例举的概念为用 Verilog 进行层次化设计提供了基础。至此，学习了如何在门级和开关级写 Verilog 代码。这两个层次可以单独运用，也可以在同一个模块中互相混合使用。我们将用这两个建模层次作为工具，学习层次化设计的基础。

让我们从一个简单的例子开始。假设运用图 10.13 中所示的电路建立了 NAND2 和 NOR2 门的开关级模型。它们是用 Verilog 模块来描写的，分别命名为 fet_nand 和 fet_nor2。我们的目的是通过例举这两种门来构成一个 AND4 门的模块。图 10.18 表示它的逻辑方块图；AND4 操作的形成很容易通过移动小圆圈的方法来验证。让我们通过例举开关级的模块来构造这个门的 Verilog 模块。

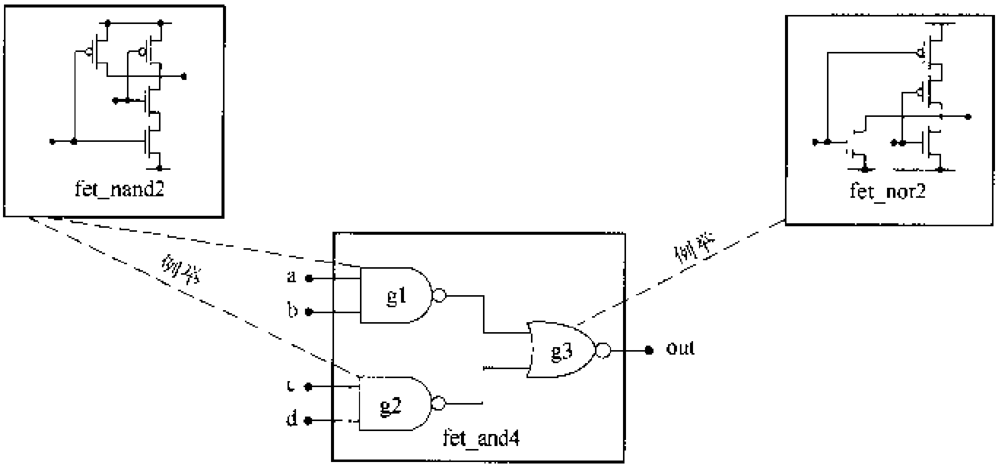


图 10.18　构建一个 AND4 门的模块

```
module fet_and2(out,a,b,c,d);
  input a,b,c,d;
  output out;
  wire out_nor,out_nand1,out_nand2;
  //门的例举 Gate instances
  fet_nand2 g1(out_nand1,a,b),
            g2(out_nand2,c,d);
  fet_nor2 g3(out,out_nand1,out_nand2);
endmodule
  /* nand 和 nor 模块必须列出在整个代码中
  以确保它们通过例举而被定义 */
```

这说明了如何进行例举的步骤,它假设模块 fet_nand2 和 fet_nor2 已用前面写好的模块定义。假设想用 fet_and4 模块构成更复杂的电路,将称这一新的模块为 group_1,它可以利用已经定义的任何实体来构成。图 10.19 显示这个单元如何例举开关级模块和 fet_and4 模块(虚线)以及联合 Verilog 基本单元 XOR 门来构造。这一模块的基本特点可以由以下的形式来概括:

```
module group_1(out_group_1,...);
  ...//输入和布线说明
  output out_group_1;
    //门例举
  fet_and4(...);
  fet_nor2(...);
  xor(...);
endmodule
```

它显示了各个层次和基本单元(**xor**)的混合使用。自然新的 group_1 模块本身又可以在下一个较高的层次上被例举。依次类推,这一类步骤可用结构化的方式来设计 VLSI 开关和逻辑电路并且可以形成文档和进行搜索。它同时简化了验证,这是因为错误常常可以通过它们在一定模块中发生的情形而比较容易定位,而且在 HDL 描述以及在物理设计阶段单元库的应用之间存在一一对应的关系。

让我们考虑在 VLSI 系统层次上遇到的问题。由于不可能跟踪通过一个复杂系统的每一位,必须转向较高层次的建模。这会增加反映结构特点所必须的观点和编码的抽象程度。假

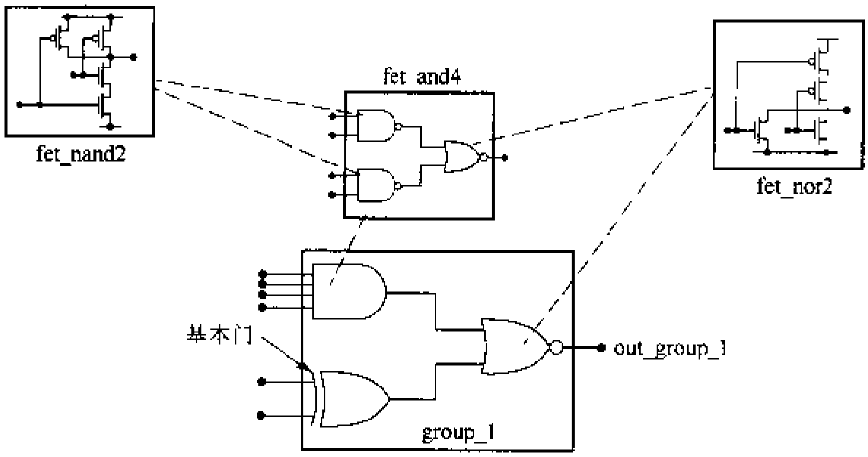


图 10.19 建立更高一层的单元

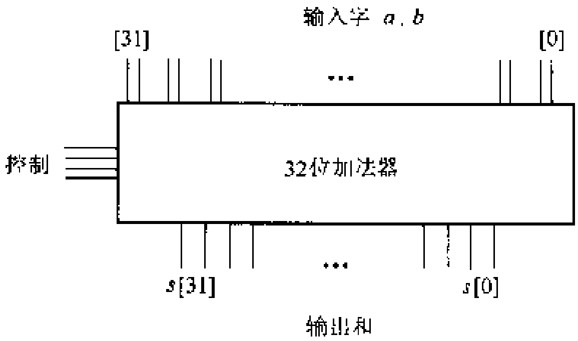


图 10.20 32 位加法器的功能方块图

设需要在的设计中包括一个 32 位的加法器。在结构级，一个模块的重要特点是它所完成的功能以及延时和时序特点，因为这些对于它与其他模块的接口是很关键的。就图 10.20 所示的功能块，我们会集中在这样一些说明上，如字长(32 位)，输入(a 和 b)和输出(s)以及运用的任何控制信号(以指出例如带符号的还是无符号的加法)。在这一层次上对模块的内部细节并不十分感兴趣；在设计中运用单元时并不需要知道电路是如何产生结果的。自然，如果希望实际构成加法器，那么电路是很重要的。

现代的 VLSI 系统设计起始于顶部的结构层并且向下一直到物理层，这是因为必须首先保证设计是正确的，然后才去关心硅上的多边形。正如前面提到的，这称为**自上而下**设计。从本质上假设了能在硅上建造所需要的单元，并且使它们接口在一起以满足系统的要求。经验是规划硅片面积和速度限制及与结构关系的最好指南。随着芯片复杂度的提高，这已变得更加困难。幸运的是硅工艺和 CAD 工具每年都在改进。

HDL 通过引入不同层次的抽象，提供系统级设计的强大工具。最高的 Verilog 层次称为**行为模型**。正如其名，它集中描述单元的整体行为，以说明它们在嵌入较大系统中时将如何工作。在行为模型中，时序常常是最关键的特点，但并不说明单元的内部细节，这些细节也不影响建模。这里假设这些说明是实际上可实现的内部电路的结果。

向下的下一个抽象层次通常称为寄存器传输级(**RTL**)模型。RTL 集中说明数据在硬件部分之间的移动。其名本身来源于这样一个事实，即同步的数字系统很大程度上依赖于运用时钟控制的存储寄存器。数据传输发生在时钟规定的特定时刻。一个 RTL 描述可以看成在纯抽象模型和硬件设计之间的联系。RTL 代码常常是产生门级网表的设计综合阶段的输入(见图 10.1)。

本章的其余章节介绍 Verilog 的高层次行为模型。这一部分内容包括用简短的例子来说明行为和 RTL 编码的基础，说清楚结构和概念。对于特定应用的比较高级的结构和编码技术

将在以后几章中介绍。

10.5 行为级和 RTL 建模

Verilog 的行为模型基于说明一组表示一个功能块特点的并行过程,重点在于精确表示结构而忽略大多数的实现细节。这一特点使编码风格非常抽象。

行为级模型的基础是构造**过程(程序)块**。正如其名,一个过程块列出描写一组操作如何执行的语句。其中的许多类似于 C 编程语言中的结构,但它们对设计过程引入了新的抽象层次。过程块包含赋值语句,高层次的结构,如循环和条件语句,以及时序控制。有两类过程块,它们起始于关键词 **initial** 和 **always**。一个 **initial** 块在模拟中执行一次,用来建立初始条件及一步步的数据流动。一个 **always** 块在模拟期间循环执行和重复。过程块把两个或更多的语句组合在一起,顺序执行的语句则插在关键词 **begin** 和 **end** 之间。另外,运用 **fork** 和 **join** 关键词也可以写并行执行的语句。

让我们从写一个时钟变量 clk 的模块开始。假设时钟周期是 10 个时间单元,所以这个变量每 5 个时间单元就改变一次,如图 10.21 所示。

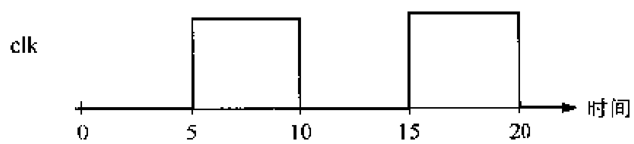


图 10.21 时钟波形 clk

```
module clock;
reg clk;
//下一条语句使时钟开始,即在 t=0 时其值为 0
    initial
        clk=1'b0;
//当在过程块中只有一条语句时,不需要进行任何组合
    always
        #5clk= ~clk;
    initial
        #500 $ finish; //模拟结束
endmodule
```

周期变化是用语句中的 NOT 算子"~"得到:

```
#5clk= ~clk;
```

因为它在 **always** 语句中,所以这个命令循环执行,直到 500 个时间单位模拟结束时为止。

1. 算子(Operator)

Verilog 算子如"~"等总结在图 10.22 中供参考。注意某些符号如"&"取决于上下文而有不同的应用。我们将学习几个,以理解它们是如何工作的。

首先考虑一元操作(也就是对单个操作数进行操作)或归并操作(reduction)。假设对 a 和 b 进行二进制赋值使 a=1101 和 b=0000。"按位求反"的操作"~"是使 a 和 b 变为:

~a=0010
~b=1111

运算	
+	加
-	减
*	乘
/	除
%	模运算

归并（一元操作）	
&	归并与
|	归并或
~&	归并与非
~|	归并或非
^	归并异或
^~	归并异或非
~^	归并异或非

移位操作	
>>	右移
<<	左移

按位操作	
~	非
&	与
~&	与非
	或
~|	或非
^	异或
~^	异或非
^~	异或非

比较关系与逻辑	
>	大于
>=	大于或等于
<	小于
<=	小于或等于
!	逻辑值求反
&&	逻辑与
||	逻辑或
==	相等
!=	逻辑不等
===	情况相等
!==	情况不等

图 10.22 Verilog 算子

即它对每位单独进行操作。“逻辑求反”的操作是使 a 和 b 变为：

! a=0
! b=1

即逻辑算子!A 得到 A 的逻辑值的反。它的含义如下：假设 A 的所有位是 0，那么它的逻辑值就是“伪”(0)。假设它不为 0，那么它的逻辑值就是“真”(1)；而!A 则给出 A 的逻辑值的反。归并算子对一个数的每位进行操作并且得到单个位的逻辑值“真”(1)或“伪”(0)。例如，当 a 和 b 如前所述定义时

&a=0
&b=0
|a=1
|b=0
^a=1
^b=0

用于 OR 的符号“|”称为“**管道**”(pipe)操作。

下面一组是二操作数操作，即它具有两个操作数。它们可以进行按位操作和逻辑操作。当 a=1010 和 b=0011 时，这些算子的按位操作是按一位一位的方式进行的：

a & b=0010
a|b=1011
a^b=1001

而进行逻辑操作时，其结果是单个的逻辑值“真”(非“零”)或“伪”(全为零)：

a && b=1
a || b=1
a && c=0

这里 c=0000。

"等于"算子为"＝"，"＝＝"及"＝＝＝"。其中赋值算子"＝"把一个表达式右边的值复制到左边，如

```
a=4'b1010
```

"等于"算子"＝＝"

```
a==b
```

用来表示"a 等于 b"。而"恒等"算子为：

```
c===d
```

它表示"c 恒等于 d"。

2. **时间控制(Timing Contral)**

时间控制语句确定动作发生的时间。在过程程序块中有三种类型的时间控制，与时钟的例子一样，简单延时用＃＜time＞说明。一个边沿触发控制的形式是@(signal)。在下面语句中

```
@(posedge clk)reg_1=reg2;
```

关键词 **posedge** 用在当时钟 clk 从 0 上升至 1 或者从 x 或 z 上升至 1 时实现赋值。图 10.23 表示了时钟的正沿和负沿。类似地，一个负沿触发的事件可用以下形式的语句来描述：

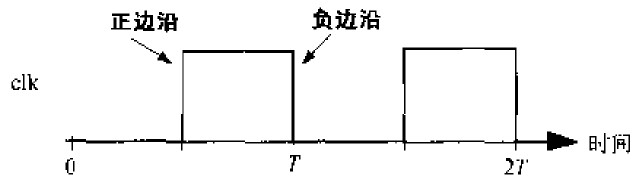


图 10.23　时钟边沿

```
@(negedge clk)output=a_in;
```

一个负沿过渡是从 1 至 0，或者从 x 或 z 至 0。边沿触发语句可以运用关键词 *or* 来包含几个信号发生变化的可能。电平触发事件用关键词 **wait** 来模拟：

```
wait(clk)q_out=d_in;
```

它在 clk 为 1 时实现这个赋值。一般地，**wait** 指令是在表达式为逻辑 TRUE(也就是非零)时执行。

3. **过程赋值(Procedural Assignments)**

过程赋值用来改变或更新 reg 和其他变量的值。它们通常分为 **blocking** 和 **non-blocking** 赋值(屏蔽赋值和非屏蔽赋值)。

屏蔽赋值按它们列出的次序执行，并且允许直接按顺序和并行的程序块。在这些语句中运用赋值算子"＝"。考虑如下简单代码：

```
reg a,b,c,reg_1,reg_2;
```

```
initial
begin
  a=1;
  b=0;
  c=1;
  # 10 reg_1=1'b0;
  # 5 c=reg_1;
  # 5 reg_2=b;
end
```

这些语句的顺序是很重要的。a,b 和 c 的赋值同时在时间 0 时执行。在 10 个时间单位之后，reg_1 赋以二进制值 1。然后在这个事件发生 5 个时间单位后(即在第 15 个时间单位时)c 赋以 reg_1 的值。最后在 20 个时间单位处，执行 reg_2 = b。图 10.24 总结了这些事件。

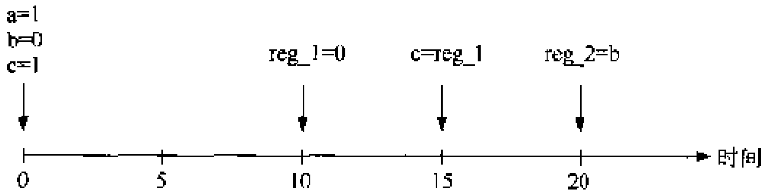


图 10.24 非屏蔽赋值的例子

非屏蔽赋值包含<=赋值算子，不要将它同“小于或等于”的关系算子相混淆。一个非屏蔽语句并不屏蔽(阻止)所列出的其他语句执行。在以下代码中：

```
A <= input_a;
B <= input_a & input_b;
```

式子右边按照任何时序控制语句首先赋值；然后这些值被传送到左边的变量。非屏蔽赋值要求仔细考虑相关的量。考虑如下的编码：

```
initial
    begin
      in_a=1;
      in_b=0;
    end
always
     # 10 clk = ~clk;
always @(posedge clk)
    begin
      in_a <= in_b;
    in_b <= in_a;
  end
```

它的作用是在每个时钟上升边沿，in_a 和 in_b 将它们原先的赋值互换。

4. **条件语句**(Conditional Statements)

取决于条件的语句在行为模型中是功能很强的结构。它们存在在最高抽象层次且用类似于 C 的语句组合描写事件。

if/else-if 结构取决于当前的条件允许有不同的结果。让我们来考察以下列出的模块：

```
module if_else_example(ctrl,alu_op,clk);
input[1:0]alu_op;
input clk;
oputut ctrl_a;
reg ctrl_a;
always @(posedge clk)
     begin
       if(alu_op==0)
            ctrl=0;
       else if(alu_op==1)
            ctrl=1;
       else
            $display("Signal ctrl is greater than 1");
       end
endmodule
```

它确定在时钟 clk 的正沿处 alu_op 的值,并根据 ctrl 给出三种可能的结果。最后一个 **else** 行则考虑了 alu_op 未列出的值即 alu_op=2,3。**If/else** 结构是可以嵌套的,其中 **else** 与最近的 **if** 相关。

case 语句则是另一个功能很强的结构。它具有如下形式:

```
case(condition)
```

它根据条件值确定结果。这从下面简单的 2:1 多路选择器的描述中可以看出:

```
module simple_mux(mux_out,p0,p1,select);
input p0,p1;
input select;
output mux_out;
always @(select or p0 or p1)
   case(select);
   1'b0:mux_out=p0;
   1'b1:mux_out=p1;
   endcase
endmodule
```

case 列出了 select 两种可能的值,并且确定无论何时在 **always** @语句中列出的变量发生变化时,哪个输入 p0 或 p1 送往 mux_out。

另一类代码由循环语句得到。**repeat** 循环使一组语句执行一定次数。假设变量 counter 的值为 10,那么,

```
repeat(counter)
     begin

       ...
     end
```

循环执行在 **begin/end** 语句之间列出的过程,总共执行 counter=10 次。另一个与循环相关的关键词为 **while**,它的句法如下:

```
while(condition)
   begin
```

```
    ...
  end
```

它执行 **begin/end** 语句块直到 condition 为"真"(非零)。如果 condition 最初为"伪"(零),那么整个语句块就被忽略(不执行)。Verilog 也有 **for** 结构,其句法为:

```
for(条件)...
```

它可测试是否执行语句块的条件表达式。

forever 循环确实如其名,它在全部模拟时间上执行。以下的时钟产生模块说明这个结构:

```
module clk_1;
reg clk;
initial
   begin
     clk=0;
     forever
        begin
        #5 clk=1;
        #5 clk=0;
        end
   end
endmodule
```

还有一些其他的条件结构,但以上这些说明了最常用的编码特点。

5. 数据流模型和 RTL(Dataflow Modeling and RTL)

数据流模型描述一个系统的数据是如何移动和处理的。与一般的行为模型一样,一个数据流描述一个高层次抽象而不提供结构细节。尽管定义有可能不同,但寄存器传输级(RTL)模型通常被看成数据流和行为编码风格的集合。它运用高层次的结构,这些结构可以作为一种综合工具的输入,后者用来产生一个门级网表。并不是所有行为级的关键词和语句都是可以被综合的,所以 RTL 大致集中在有限的一组语句。掌握 RTL 通常需要反复学习如何写可综合的代码,但这已远超出本书的范围。

我们将限于介绍关键词 **assign**。连续赋值可以定义逻辑关系和数值。例如以下形式的语句

```
assign a=~b & c;
assign out_1=(a|b)&(c|d);
```

可以用来定义组合逻辑操作。一个很有用的条件语句是:

```
assign output=(something)? <true condition>:<false condition>
```

在这种情形中,something 代表一个变量或一条语句。output 的值取决于 something 是真还是伪。一个 2:1 MUX 的描述可以写成:

```
module mux_2(out,p0,p1,select);
input p0,p1;
input select;
output mux_out;
```

```
assign out = (select)? p1 : p0;
endmodule
```

这类语句可以嵌套。在后面几章中将会看到更多有关 **assign** 和其他数据流结构的例子。

10.6 参考资料

Verilog 是一个丰富和功能很强的语言，它有许多复杂的细节。下面列出的所有书籍都提供比本书所能包括的更为深刻的内容。参考资料[3]是一本教科书，而[2]和[8]则写成教科书的风格，但只有较少的例子和细节。参考资料[7]快速简明介绍 Verilog。而[9]和[12]是一本较全面的参考书。

[1] Mark Gordon Arnold, **Verilog Digital Computer Design**, Prentice-Hall PTR, Upper Saddle River, NJ, 1999.

[2] J. Bhasker, **A Verilog HDL® Primer**, Star Galaxy Press, Allentown, PA, 1997.

[3] Michael D. Ciletti, **Modeling, Synthesis and Rapid Prototyping with the Verilog HDL**, Prentice-Hall, Upper Saddle River, NJ, 1999.

[4] Ken Coffman, **Real World FPGA Design with Verilog**, Prentice-Hall PTR, Upper Saddle River, NJ, 2000.

[5] Dan Fitzpatrick and Ira Miller, **Analog Behavioral Modeling with the Verilog-A Language**, Kluwer Academic Press, Norwell, MA, 1999.

[6] Pran Kurup and Taher Abasi, **Logic Synthesis Using Synopsys®**, 2nd ed., Kluwer Academic Publishers, Norwell, MA, 1997.

[7] James M. Lee, **Verilog Quickstart!**, Kluwer Academic Publishers, Norwell, MA, 1998.

[8] Samir Palnitkar, **Verilog® HDL**, SunSoft Press (Prentice-Hall), Mountain View, CA, 1996.

[9] Vivek Sagdeo, **The Complete Verilog Book**, Kluwer Academic Publishers, Norwell, MA, 1998.

[10] Bruce Shrive and Bennett Smith, **The Anatomy of a High-Performance Microprocessor**, IEEE Computer Society Press, Los Alamitos, CA, 1998.

[11] David R. Smith and Paul D. Franzon, **Verilog Styles for Synthesis of Digital Systems**, Prentice-Hall, Upper Saddle River, NJ, 2000.

[12] Donald E. Thomas and Philip R. Moorby, **The Verilog® Hardware Description Language**, 4th ed., Kluwer Academic Press, Norwell, MA, 1998.

[13] Bob Zeidman, **Verilog Designer's Library**, Prentice-Hall PTR, Upper Saddle River, NJ, 1999.

10.7 习题

[**10.1**] 对图 P10.1 所示的模块,写出门级结构描述。

[**10.2**] 考虑图 P10.2 所示的逻辑电路。写出描述这一电路的 Verilog 模块。

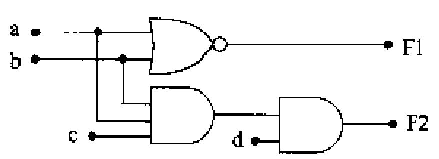


图 P10.1

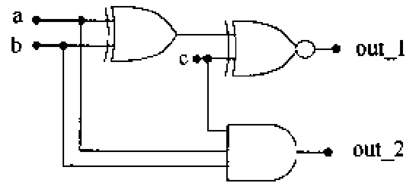


图 P10.2

[**10.3**] 写出图 P10.3 中的 NAND 锁存器的 Verilog 描述。每个 NAND 门包括一个 2 个单位的时延。

[**10.4**] 写出图 P10.4 中所示的逻辑电路的 Verilog 模块。假设 NOT 门的时延为一个单位,而 AND2 门的时延为 2 个单位。

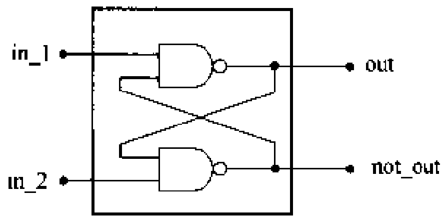


图 P10.3

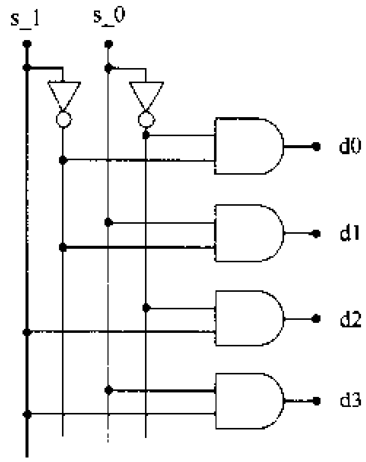


图 P10.4

[**10.5**] 画出实现以下功能的 CMOS 逻辑门的电路图。

$$f = \overline{a\cdot(b+c)+b\cdot d} \tag{10.3}$$

然后运用 **nmos** 和 **pmos** 基本单元写出这个电路的 Verilog 描述。

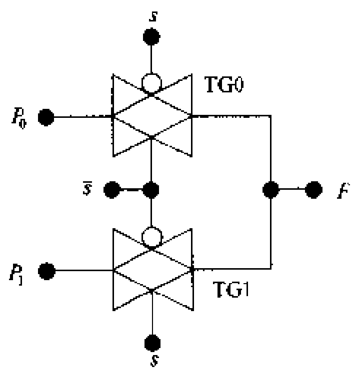


图 P10.5

[10.6] 设计一个准 nMOS 门实现以下功能:

$$F = \overline{a\cdot b\cdot c+a\cdot(d+e)} \tag{10.4}$$

然后用 **nmos** 和 **pullup** 基本单元写出该电路的 Verilog 描述。

[**10.7**] 用 cmos 基本单元写出图 P10.5 中 2:1 MUX 的 Verilog 模块代码。设每个传输门的时延为 2 个单位。

[**10.8**] 一个同步系统采用互补的时钟相位 ϕ 和 $\bar{\phi}$ 来控制数据流。写出一个 Verilog 模块的代码,实现两个不重叠的时钟信号 clk 和 clk _ bar,周期为 40 个时间单位。

［**10.9**］ 一个组合逻辑电路的输入为 a,b,c,e 和 f,实现如下功能:

$$A = \bar{a} \cdot b + c \cdot e + a \cdot (\bar{c} + f) \tag{10.5}$$

将结果 A 连到一个正沿触发的 DFF 的输入端。DFF 的输出为 q。采用合适的一组基本单元写出这个电路的 Verilog 描述。

第 11 章　常用的 VLSI 系统部件

VLSI 系统设计要认真考虑 -个部件功能库。基本的实体包括 FET 以及基本的逻辑门，但还需要较高层次的功能以建立起系统层次体系。本章中将研究用来构成大规模系统和在 VLSI 单元库中通常遇到的系统部件的几个例子。这里列出的单元并不是很全面的，另外的部件将在后面几章中介绍。然而我们的步骤意在强调高层次体系结构的描述与所生成电路及硅片实现之间的联系。

11.1　多路选择器

在现代数字设计中总会用到多路选择器。一个多路选择器(MUX)包括 n 个输入线和一个输出 f。这一部件的主要功能是通过一个 m 位的选择字把其中一个输入选送到输出端。为了包括所有的输入，必须选择 m 使 $n = 2^m$。另一种说明这一部件的方式是用基数为 2 的对数，即 $\log_2(2) = 1$。于是由

$$m = \log_2(2^m) \tag{11.1}$$

可以得到

$$m = \log_2(n) \tag{11.2}$$

它就是选择线的数目。

最简单的例子是 2 至 1 的多路选择器。可以有几种方式来描述这个部件。在上一章中介绍了用 **case** 语句对它进行行为描述，这里再次列出以供参考。输入线表示为 p_0 和 p_1，而选择位由标识符 select 来表示。

```
module simple_mux(mux_out,p0,p1,select);
input p0,p1;
input select;
output mux_out;
always @(select)
    case(select)
    1'b0:mux_out = p0;
    1'b1:mux_out = p1;
    endcase
endmodule
```

根据这一描述，可以得到几个不同的逻辑和电路实现。图 11.1 为门级 NAND 的实现。运用 DeMorgn 定理，通过小圆圈的推移法可以得到 SOP(AND - OR)的形式(积项之和的形式)：

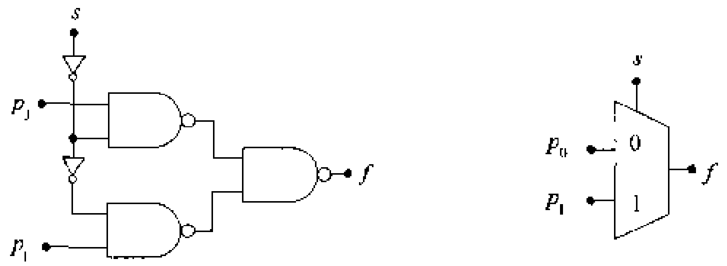
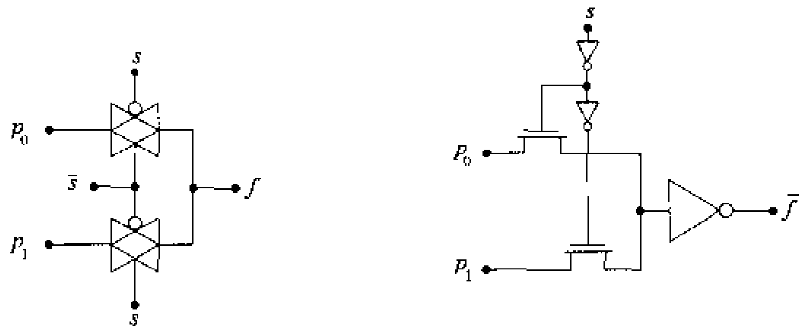


图 11.1　门级 NAND 2:1 多路选择器

$$f = p_0 \cdot \bar{s} + p_1 \cdot s \tag{11.3}$$

由于 NAND2 门要求 4 个 FET，因此带有驱动器的整个电路可以用 16 个晶体管来实现。①

图 11.2(a)的传输门电路也可作为 CMOS 工艺实现的一种选择。这一电路运用 4 个 FET 实现路径选择逻辑(每个 TG 有两个管子)。如果包括一对用于选择位的缓冲 NOT 门，那么总的 FET 数目增加到 8 个。这个电路的主要问题是 TG 具有寄生电阻和电容，会使响应减慢。图 11.2(b)为只用 nFET 开关的相同电路形式。包括选择线驱动器在内，FET 的数目减少到 6 个。但在输出端增加了一个反相驱动器以补偿 nFET 只能通过电压范围为 0V 至 V_{max}的事实：



(a) TG 电路　　(b) 传输管

图 11.2　用开关逻辑实现多路选择器

$$V_{max} = V_{DD} - V_{Tn} \tag{11.4}$$

式中 V_{Tn}是阈值电压。这一反相器帮助把输出恢复至全幅的电压范围$[0, V_{DD}]$。尽管现在 FET 的数目与用 TG 电路时相同，但版图的布线将比较容易。当把高层次描述转变成硅片时，像这样一些考虑就成为很重要的因素。

较大的多路选择器可以用基本门或例举 2:1MUX 器件来设计。考虑以下描述的 4:1MUX：

```
module bigger_mux(out_4,p0,p1,p2,p3,s0,s1);
input p0,p1,p2,p3;
input s0,s1;
output out_4;
assign out_4 = s1 ? (s0? p3:p2):(s0? p1:p0);
endmodule
```

① 这个电路也可以用结构代码来描述。

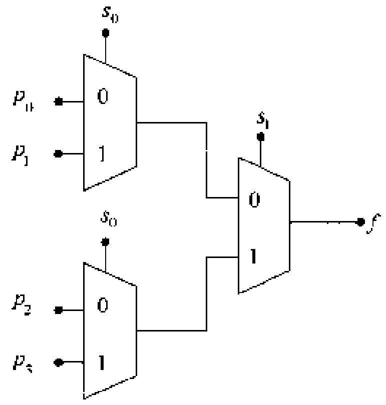
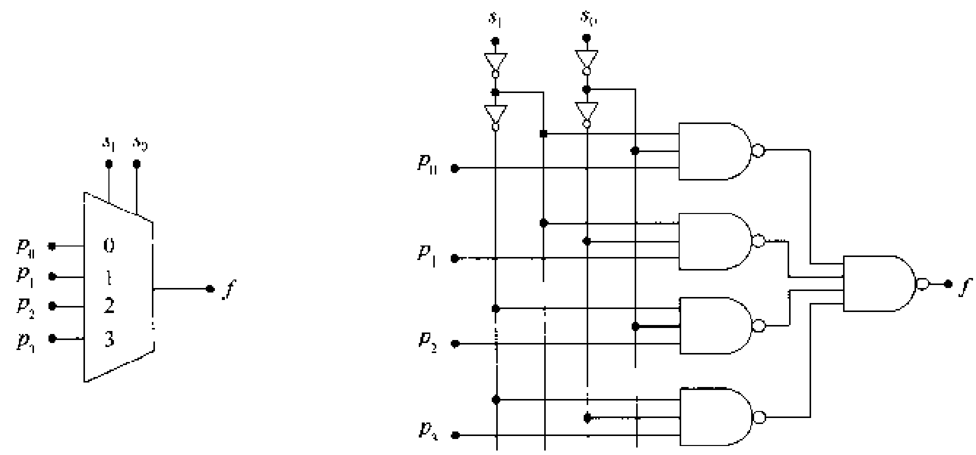


图 11.3 用例举 2:1 MUX 器件实现 4:1 MUX

由于这是一个高层次的抽象描述，没有给出有关内部结构的任何细节。然而以上描述中的赋值语句可以看做三个单独的2:1多路选择器，其中第一个? 表示用 s_1 来选择，而第二和第三个? 则基于 s_0 来选择，这相当于图 11.3 所示的结构。在第一级器件中选择位 s_0 用来选择(p_0, p_2)或(p_1, p_3)。最终的选择由 s_1 来完成，它决定了实际的输出 f，这与上面列出的 Verilog 描述中的 out_4 相同。

另一种实现是图 11.4 所示的门级结构。以此图为基础的等效 Verilog 结构描述如下：

```
module gate_mux_4(f,p0,p1,p2,p3,s0,s1);
input p0,p1,p2,p3;
input s0,s1;
wire w1,w2,w3,w4;
output f;
nand(w1,p0,~s1,~s0),
     (w2,p1,~s1,s0),
     (w3,p2,s1,~s0),
     (w4,p3,s1,s0),
     (f,w1,w2,w3,w4);
endmodule
```



(a) 符号 (b) 逻辑图

图 11.4 门级 4:1 MUX

NOT 门是用~算子来模拟的，但它们也可以用基本的 not 门例举来得到相同的结果。采用标准逻辑时，这相当于以上的 SOP 表达式(积项之和的形式)：

$$f = p_0 \cdot \overline{s_1} \cdot \overline{s_0} + p_1 \cdot \overline{s_1} \cdot s_0 + p_2 \cdot s_1 \cdot \overline{s_0} + p_3 \cdot s_1 \cdot s_0 \qquad (11.5)$$

上式可以应用基本逻辑得到。

还有另一种电路是图 11.5 中的传输 FET 阵列。它利用 nFET 的"与"性质直接实现逻辑表达式。这一电路的结构描述如下：

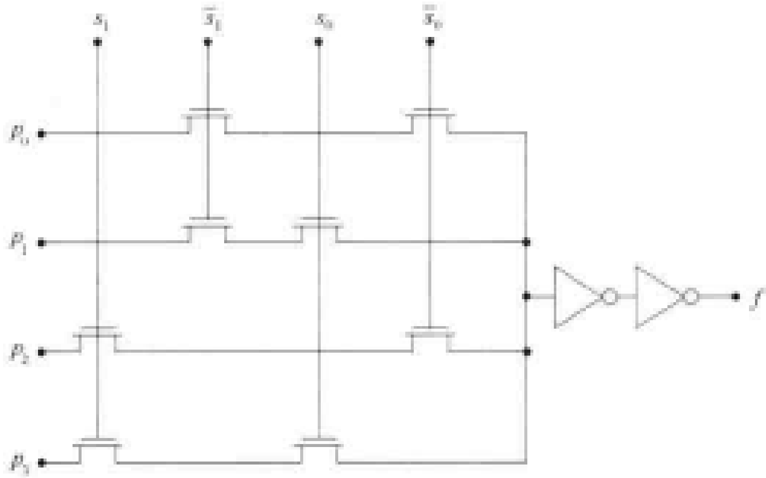


图 11.5　用 nFET 传输管实现 4:1 MUX

```
module tg_mux_4(f,p0,p1,p2,p3,s0,s1);
input p0,p1,p2,p3;
input s0,s1;
wire w0,w1,w2,w3,w_o,w_x;
output f;
nmos(p0,w0,~s1),(w0,w_o,~s0);
nmos(p1,w1,~s1),(w1,w_o,s0);
nmos(p2,w2,s1),(w2,w_o,~s0);
nmos(p3,w3,s1),(w3,w_o,s0);
not(w_x,w_o),(f,w_x);
endmodule
```

这里 **nmos** 举例已被组合在一起以便查找。第 i 行中 FET 之间的连线标记为 w_i，其中 $i=0,1,2,3$，而 w_o 是这四条路径的公共输出线。在反相器之间的连线为 w_x。图 11.6 为这个单元简单的版图画法。这里一一对应的关系十分明显，然而通过移动 FET 和重新布线可以提高集成密度。

图 11.7 分成两个阵列的 nMOS/pMOS 电路运用类似的道理提供同一功能。每个输入同时有两条路径分别经过 nFET 和 pFET 链通向输出端。由于 pFET 可以通过逻辑 1 电压而 nFET 可以通过逻辑 0 电压，因此输出具有 0 V 至 V_{DD}的全摆幅。在这一情形中不需要输出电平恢复缓冲器，倘若加入缓冲器则该电路可以快得多。这一电路采用了与 TG 连线方式类似的互补对概念。然而由于 nFET 和 pFET 电路是隔开的，因此互连线布线比起每个开关采用 TG 要简单。这些例子表明在将高层次 HDL 结构转变为一个基本的逻辑电路时可以有几种不同的实现方式。

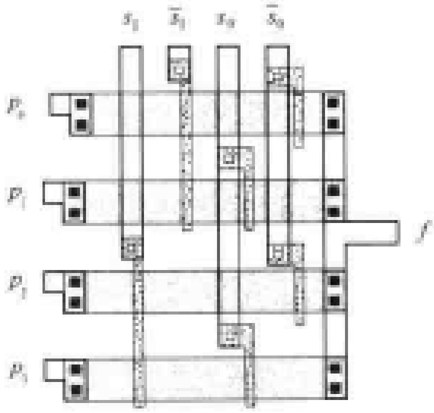


图 11.6　4:1 传输 FET MUX 的简单版图

结构描述对于 n 位字采用与一位相同的方式进行。假设我们有两个 8 位的字：

$$\begin{aligned} a &= a_7a_6a_5a_4a_3a_2a_1a_0 \\ b &= b_7b_6b_5b_4b_3b_2b_1b_0 \end{aligned} \tag{11.6}$$

希望把它们作为一个 2:1 MUX 的输入。其输出

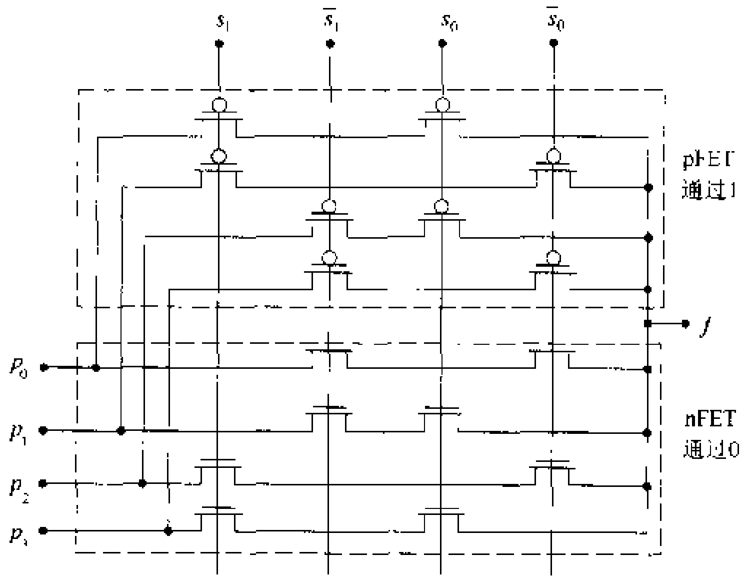


图 11.7 具有全摆幅输出分开阵列的 4:1 MUX

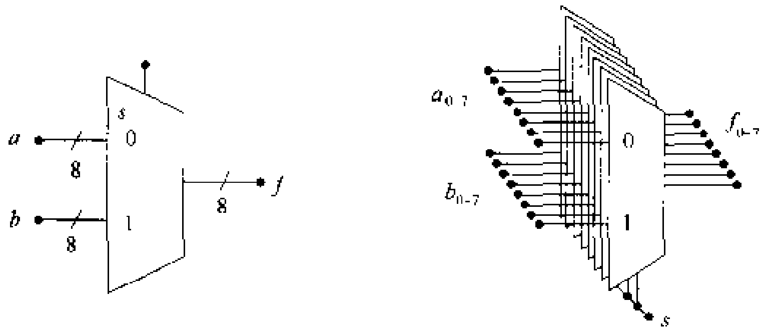
$$f = f_7 f_6 f_5 f_4 f_3 f_2 f_1 f_0 \tag{11.7}$$

是由选择位 s 来决定的：

$$f_i = a_i \cdot \bar{s} + b_i \cdot s \tag{11.8}$$

式中 $i=0,\cdots 7$。自然这意味着应当采用 8 个完全相同的 2:1 MUX，它们全部由相同的选择位 s 来控制。

在系统级，希望把 a 和 b 看成单个对象。图 11.8(a)中的 MUX 符号用与输入输出线交叉的斜杠(/)表示字长。Verilog 描述可用[7:0]所表示的 8 位向量列出如下：



(a) 符号　　(b) 位级实现

图 11.8 向量表示的 2:1 MUX

```
module mux_2-1_8b(f,a,b,s);
input[7:0]a,b;
input s;
output[7:0]f;
  assign f=s ? b:a;
endmodule
```

扩展到较大的字长只需要重新规定向量的维数。但在二进制位层次上实现时是比较复杂的，

因为需要采用 n 个平行的 2:1 MUX 单元,如图 11.8(b)所示。在物理层次上,每位 MUX 都占有面积且有一组延迟时间。为了构成一个 8 位的电路,可以像铺瓦片似地把 8 个相同的单元排列起来,如图 11.9 所示。由于硅片电路(以及所有逻辑门)都是在二进制位这一层次上设计的,因此版图面积和布线可能成为限制因素。

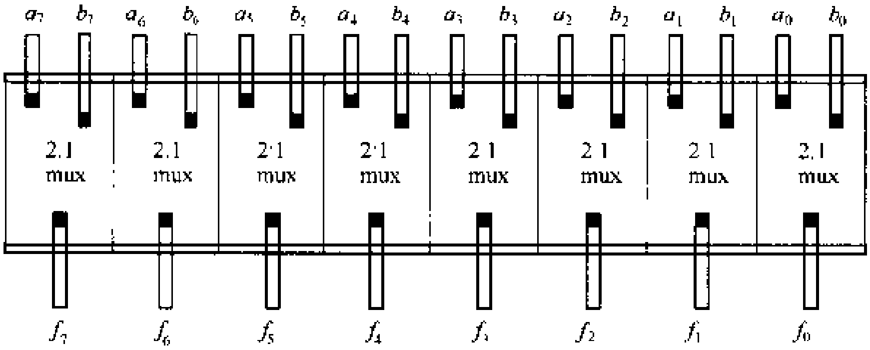


图 11.9 采用单个位的单元按铺瓦片方式构成一个 8 位 2:1 MUX

这个讨论力图说明一个重要观点。如果已有一个数字电路的高层次结构描述,那么对这个电路可以做出几种选择。每种选择都产生一个不同的物理设计,它有自己的版图和开关特性。在关键的延时路径中,TG 电路也许不够快。而在另一种情形中,一个有限的芯片资源分配也许使面积更为重要。VLSI 系统设计并不只是编写好的代码或增加一些多边形。层次化设计中的每层对其他每个层都会有某些影响。在自顶向下设计方法中,常常发现一个行为描述可以用几种方式实现,有些比其他的更好些。选择取决于在整个设计周期中所采用的衡量标准。

这个策略将贯穿在本章。将在不同的层次上分析几个逻辑部件,说明在高层次结构描述与电路或硅片电路之间的联系。虽然这些部件就它们本身而言也是很重要的,但只有在部件被用来构成更为复杂的逻辑单元之后,系统层次上的关键特点才展现出来。设计层次如图 11.10 所示的嵌套方块图表示。在自顶向下设计中,将从一组描述开始,然后用 HDL 设计高层次的结构模型。这个设计一直向下进行到硅片最终形成硅器件,但在各层次之间的相互作用则是以非常关键的方式联系在一起的。没有一个系统可以比它的每个下层可能有的速度工作得更快,而允许的硅片面积资源则是最终的限制。而且很自然,我们必须制造出芯片并以用户愿意支付的价格出售而仍能保持赢利的余地!

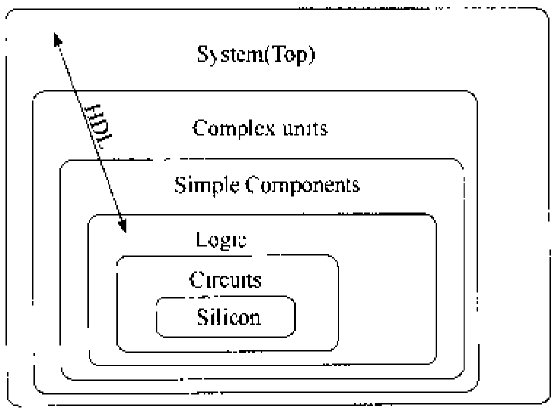


图 11.10 系统设计层次

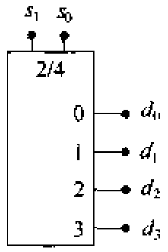
11.2 二进制译码器

一个 n/m 行二进制译码器输入 n 位的控制字,使 m 条输出线中的一条有效而其余 $(m-1)$ 条输出线不受影响。一个**高电平有效**的译码器使被选输出线置“1”而保持其余输出线为“0”。一个**低电平有效**的译码器则恰好与此相反,被选输出线置 0 而其余输出线为 1。

图 11.11(a)为一个 2/4 高电平有效的译码器符号和功能表。两位的选择字 s_1s_0 使相应于它所说明的十进制值 0,1,2,3 的输出线变为有效。由功能表得到以下方程:

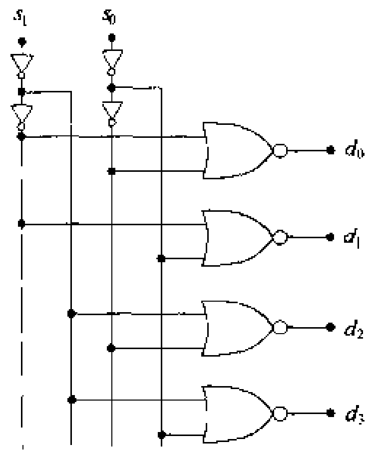
$$
\begin{aligned}
d_0 &= \overline{s_1} \cdot \overline{s_0} = \overline{s_1 + s_0} \\
d_1 &= \overline{s_1} \cdot s_0 = \overline{s_1 + \overline{s_0}} \\
d_2 &= s_1 \cdot \overline{s_0} = \overline{\overline{s_1} + s_0} \\
d_3 &= s_1 \cdot s_0 = \overline{\overline{s_1} + \overline{s_0}}
\end{aligned}
\tag{11.9}
$$

图 11.11(b)为直接采用 NOR 门的实现,它是如下结构描述的基础:



s_1	s_0	d_0	d_1	d_2	d_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(a) 符号与真值表



(b) NOR2 实现

图 11.11 高电平有效的 2/4 译码器

```
module decode_4(d0,d1,d2,d3,s0,s1);
input s0,s1;
output d0,d1,d2,d3;
nor    (d3,~s0,~s1),
       (d2,~s0,s1),
       (d1,s0,~s1),
       (d0,s0,s1);
endmodule
```

这里把反相驱动器用算子符号“~”来表示。

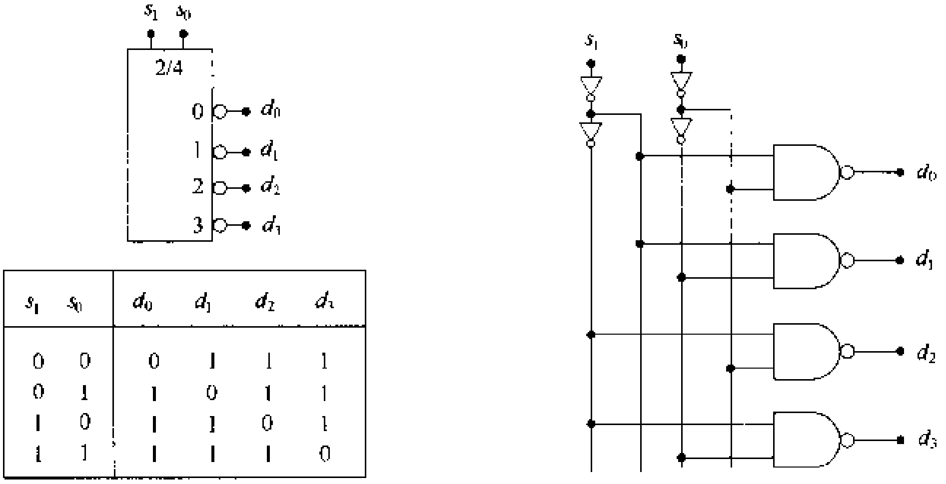
采用 **case** 关键词的等效结构描述可以写成如下:

```
module dec_4(d0,d1,d2,d3,sel);
input[1:0]sel;
output d0,d1,d2,d3;
case(sel)
  0:d0=1,d1=0,d2=0,d3=0;
  1:d0=0,d1=1,d2=0,d3=0;
  2:d0=0,d1=0,d2=1,d3=0;
  3:d0=0,d1=0,d2=0,d3=1;
endmodule
```

以上明确列出了取决于 sel 十进制值的每种可能性。另一种方法是采用 **assign** 过程。它代表

对这个操作的抽象高层次描述而不含任何结构信息。虽然可以理解这个单元的操作，但在它能被实际实现之前必须先转换成较低层次的描述。

低电平有效的译码器显示在图 11.12 中。在这一情形中，所选择的输出线被驱动至低电平而其余则保持在逻辑 1 的电平值。这一设计的实现只不过是用 NAND2 门来代替 NOR2 门，且使每个门的输入变为反信号。编写它的 HDL 代码只要根据改变的逻辑修改高电平有效译码器的代码即可。这一电路门级结构级的 Verilog 描述可以通过观察写成如下形式：



s_1	s_0	d_0	d_1	d_2	d_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

(a) 符号与真值表　　(b) NAND实现

图 11.12　低电平有效的 2/4 译码器

```
module dec_lo(d0,d1,d2,d3,s0,s1);
input s0,s1;
output d0,d1,d2,d3;
nand (d0,~s0,~s1),
     (d1,~s0,s1),
     (d2,s0,~s1),
     (d3,s0,s1);
endmodule
```

这些简单的例子清楚地表明大的部件如何在不同的层次上描述。在实际中最常用哪一个取决于所要解决的问题以及它在层次化设计中的位置。一般来说，单种解决办法不会对所有情形都是最优的。

11.3　相等检测器和比较器

一个相等检测器比较两个 n 位的字，当两个输入每位一一相等时就输出 1。一个简单的 4 位电路显示在图 11.13 中。它运用异或非(XNOR)关系：即如果有且仅有当 $a_i = b_i$ 时

$$\overline{a_i \oplus b_i} = 1 \tag{11.10}$$

上式可以用来比较两个输入。如果每个 XNOR 产生一个 1，那么输出与门就给出 Equal＝1；否则 Equal＝0。

这一操作的 Verilog 代码如下：

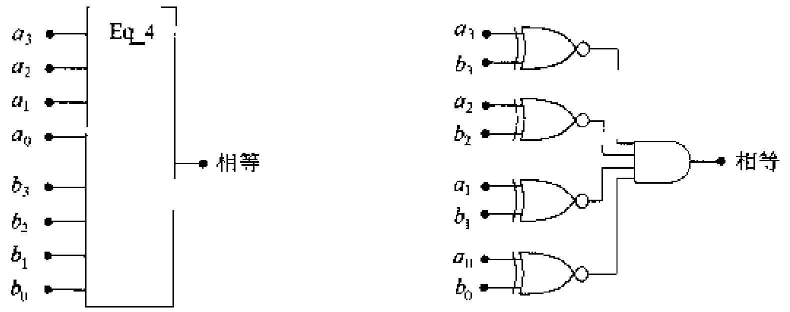


图 11.13 4 位相等检测器

```
module equality(Equal,a,b);
input[3:0]a,b;
output Equal;
always @(a or b)
    begin
      if(a==b)
        Equal=1;
      else
        Equal=0;
    end
endmodule
```

这一电路的内部结构隐含在逻辑相等 a==b 的条件中。把它扩展到任意的字长,无论在电路级还是在 HDL 级都很容易实现。图 11.14 为 8 位相等检测器的例子。它采用两个 4 位的电路并把它们的输出"与"在一起得到最终的结果。

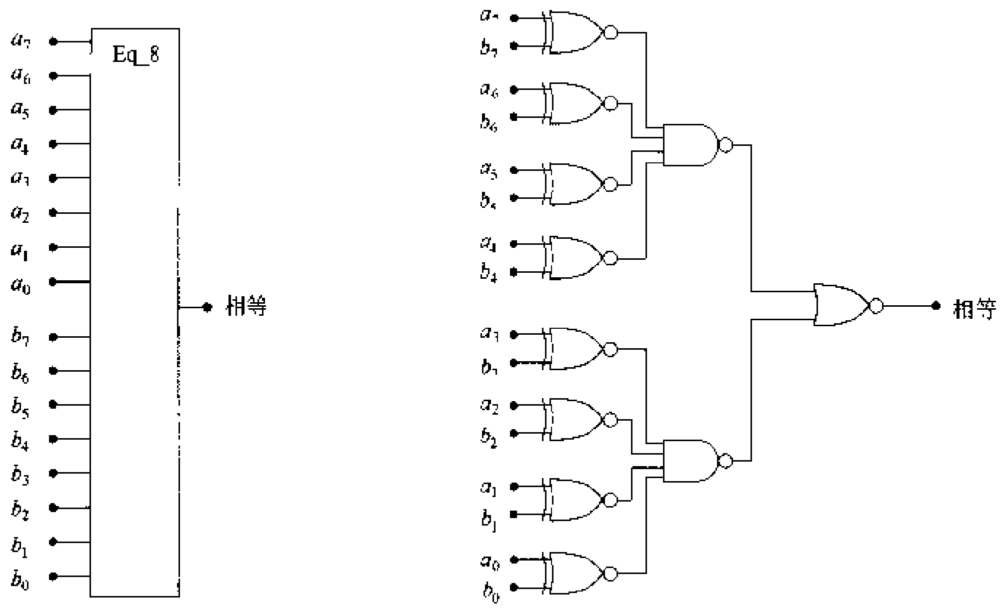


图 11.14 8 位相等检测器

数值比较器电路用来比较两个字 a 和 b,并且决定 $a>b$ 或 $a<b$ 是否成立;相等的情况即 $a=b$ 也可以用这一逻辑电路来检测。图 11.15 为一个 4 位的数值比较器。输入字按位比较最终产生两个输出 GT 和 LT,其结果总结在图 11.16 中。逻辑方程的推导有些乏味,但信号路径可以从电路图中查出来。上下两个逻辑链的对称性是产生 GT 和 LT 结果的基础。作

为一种选择可以添加一个相等检测输出和一个使能控制，即在该电路基础上再串接一个如图 11.17 所示的电路。

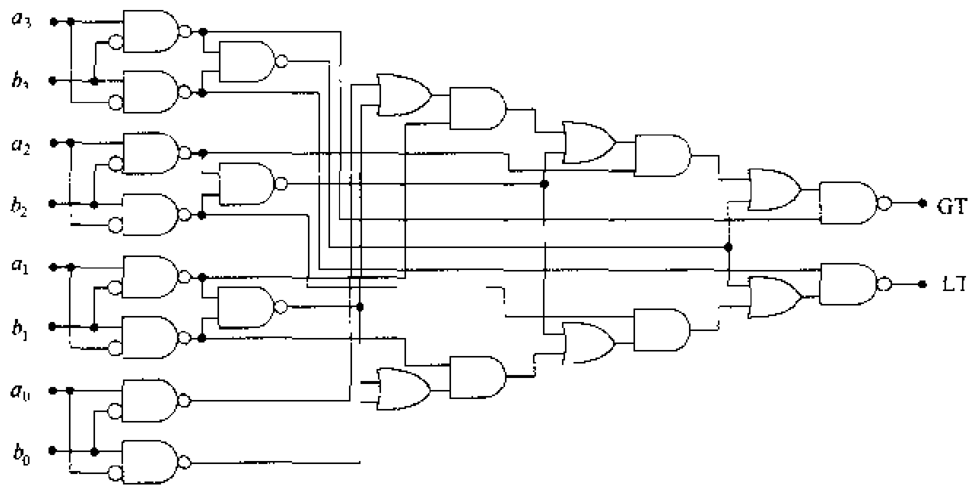


图 11.15　4 位数值比较器逻辑

条件	大于	小于
$a>b$	1	0
$a<b$	0	1
$a=b$	0	0

图 11.16　比较器输出总结

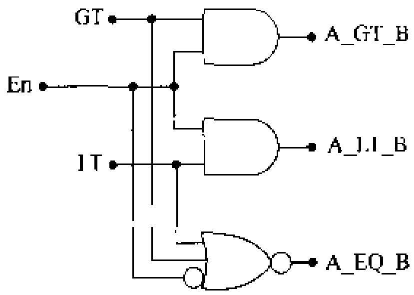


图 11.17　用附加逻辑实现 A _ EQ _ B 及使能控制

一个 4 位比较器的 Verilog 代码可以建立如下：

```
module comp _ 4(GT,LT,a,b);
input[3:0]a,b;
output GT,LT;
always @(a or b)
  begin
    if(a>b)
      GT=1,LT=0;
    elseif(a<b)
      GT=0,LT=1;
    else
      GT=0,LT=0;
  end
endmodule
```

高层次的描述完全避开内部结构，使它适合于体系结构级的模拟。然而该逻辑和电路的实现有可能是相当复杂的。

层次化设计可以用两个 4 位的比较器(Comp 4)和一个接口电路来构成一个 8 位的比较器。图 11.18 为主要电路。图中下面的 Comp 4 方块输入每个字的低 4 位，而上面的方块输入 4 至 7 位。标记 Comp 8 的接口方块包括一个使能输入。接口电路的逻辑方块图显示在图

11.19 中。上面的输入是 GT Comp 4 的输出,而下面的输入是从 4 位比较电路来的 LT 值。它们再进行比较以产生输出。包含 AND 门以及 NAND－NOT 串联电路(在 A_GT_B 和 A_LT_B 输出处)可在两个字相等时产生一个相等信号 A_EQ_B=1。注意 A_GT_B 和 A_LT_B 在 A_EQ_B=1 时均为 0。

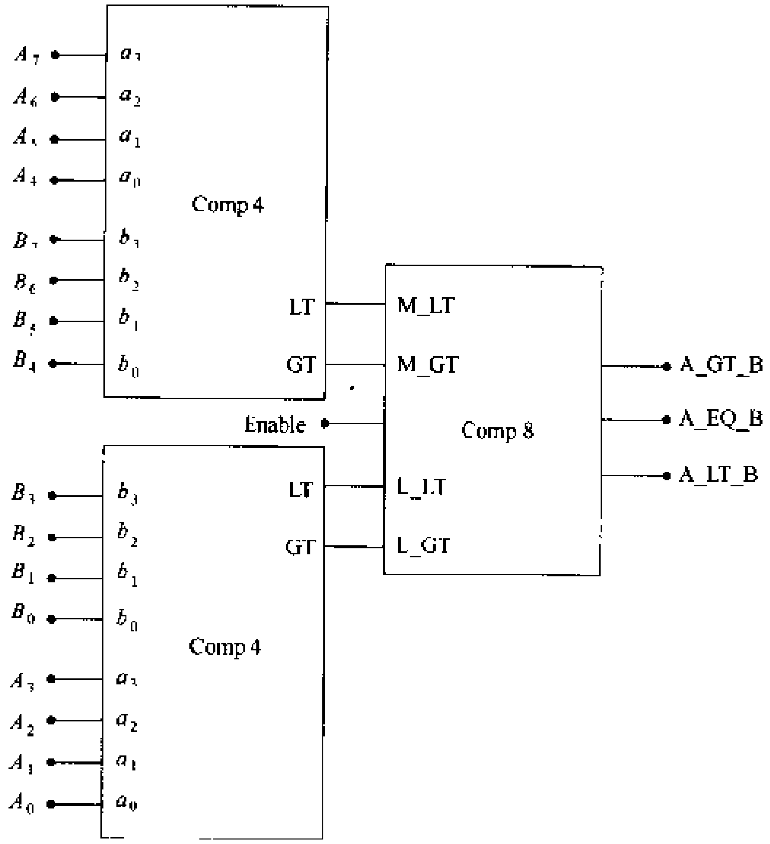


图 11.18 8 位比较器的组织

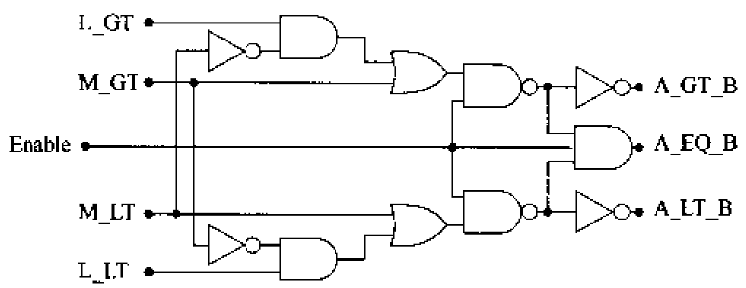


图 11.19 Comp 8 逻辑方块图

11.4 优先权编码器

一个优先权编码器考察一个 n 位字的输入位,它的输出用来表明最高优先权(逻辑值为 1)的输入位的位置。考虑一个 8 位的字

$$d = d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 \tag{11.11}$$

且把最高的优先权分配给 d_7 位，第二高的优先权分配给 d_6，依次类推。一个优先权编码器的工作是检测在 d 中存在的 1；如果两位或更多位是逻辑值 1，那么具有最高优先权的输入位占先。如果用 d 作为一个 8 位优先权编码器的输入，那么它的输出字

$$Q = Q_3Q_2Q_1Q_0 \tag{11.12}$$

则被编码以指出最高优先权位的位置。这一编码器的功能表列在图 11.20 中。如果任何一个输入位为 1，则 Q_3 位等于 1。其余三位的字 $Q_2Q_1Q_0$ 则进行编码以表明最高优先权的输入位。由于它没有正式的逻辑符号，所以当这个部件用在系统设计中时，将用图 11.21 所示的简单方块图来表示。

d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	–	1	0	0	1
0	0	0	0	0	1	–	–	1	0	1	0
0	0	0	0	1	–	–	–	1	0	1	1
0	0	0	1	–	–	–	–	1	1	0	0
0	0	1	–	–	–	–	–	1	1	0	1
0	1	–	–	–	–	–	–	1	1	1	0
1	–	–	–	–	–	–	–	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0

d_7 有最高优先权
d_0 有最低优先权

当 d_i=1 时 Q_3=1
对任何 i=0、7,

图 11.20　8 位优先权编码器的功能表

这一电路的逻辑被画成两部分。图 11.22 的第一部分显示每位的输入缓冲器和产生每位反信号的逻辑。Q_2 和 Q_3 的输出逻辑很简单，可以用下式来表示：

$$Q_2 = (d_0+d_1+d_2+d_3)\cdot\overline{(d_4+d_5+d_6+d_7)}$$
$$Q_3 = (d_0+d_1+d_2+d_3)+\overline{(d_4+d_5+d_6+d_7)} \tag{11.13}$$

它们可以从图中得到验证。Q_0 和 Q_1 编码器运用经缓冲和反相的输入，如图 11.23 的电路所示。Q_0 电路的逻辑方程为：

$$Q_0 = \overline{d_7}\cdot[d_6+\overline{d_5}\cdot(d_4+\overline{d_3}\cdot[d_d+\overline{d_1}\cdot d_0])] \tag{11.14}$$

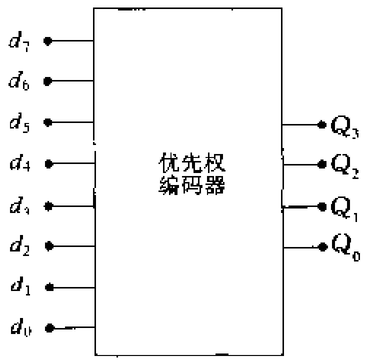


图 11.21　优先权编码器的符号

而

$$Q_1 = \overline{d_7}\cdot\overline{d_6}\cdot[d_5+\overline{d_4}+\overline{d_3}\cdot\overline{d_2}\cdot(d_1+d_0)] \tag{11.15}$$

则给出了 Q_1 位。

尽管电路内部的细节很复杂，但行为描述只考虑整个的功能行为。这一模块的一种实现为：

```
module priority_8(Q,Q3,d);
input[7:0]d;
```

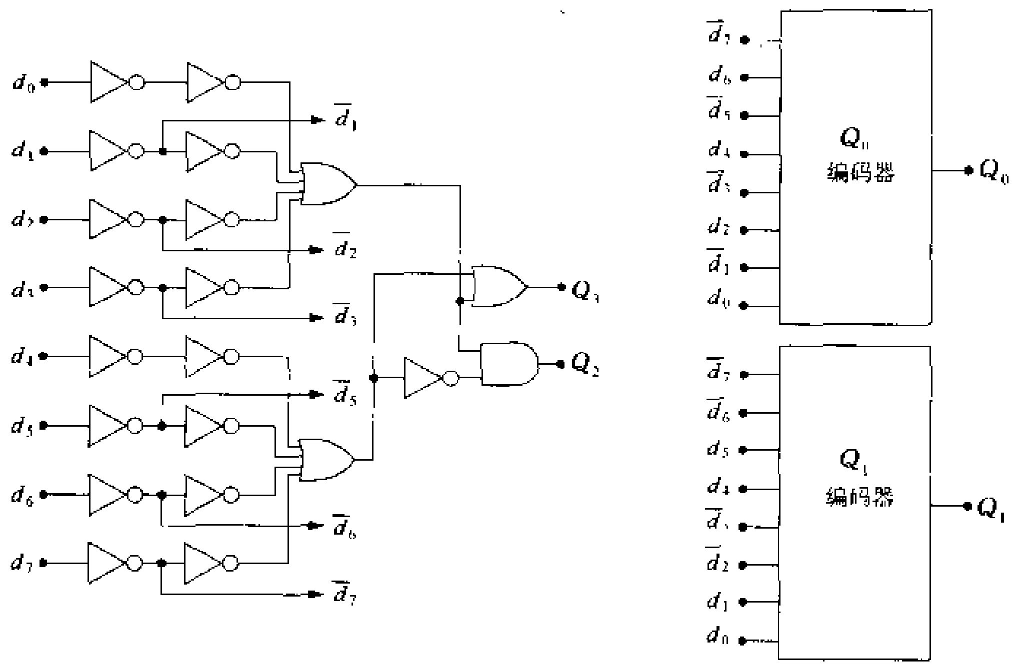
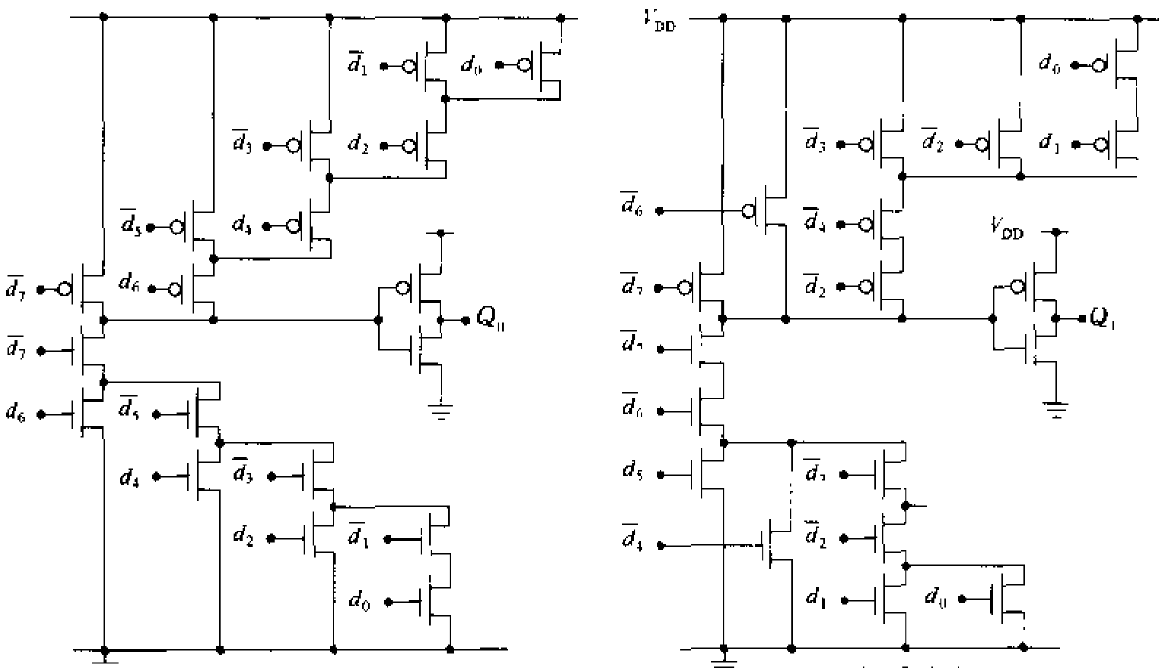


图 11.22 优先权编码器的逻辑图



(a) Q_0电路 (b) Q_1电路

图 11.23 8 位优先权编码器中的 Q_0 和 Q_1 电路

```
output Q3;
output[2:0]Q;
always @(d)
    begin
      Q3=1;
      if(A[7])Q=7;
      elseif(A[6])Q=6;
```

```
        elseif(A[5])Q=5;
        elseif(A[4])Q=4;
        elseif(A[3])Q=3;
        elseif(A[2])Q=2;
        elseif(A[1])Q=1;
        elseif(A[0])Q=0;
        else
          begin
            Q3=0;
            Q=3 'b000;
          end
      end
endmodule
```

我们定义了 Q_3 为一个标量而 Q 为一个三位的向量，Q 是根据在功能表中列出的 $Q_2Q_1Q_0$ 的等效十进制数来赋值的。这一例子非常好地说明了在高层次和低层次描述之间的差别。把 HDL 转换成电路图并不是一个简单的问题，况且还可以构成其他等效的电路和逻辑算法，每个会有不同的面积和开关特性。

11.5 移位和循环操作

移位和循环单元在许多不同的电路中非常有用。考虑一个 4 位的字 $a_3\,a_2\,a_1\,a_0$ 作为如图 11.24 所示一般循环单元的输入，其输出为经过循环后的字 $f_3\,f_2\,f_1\,f_0$。一个 n 位循环是由控制字 RO _ n 来说明的，而 L/R 位则定义了左移或右移。例如，一位左循环产生如下输出：

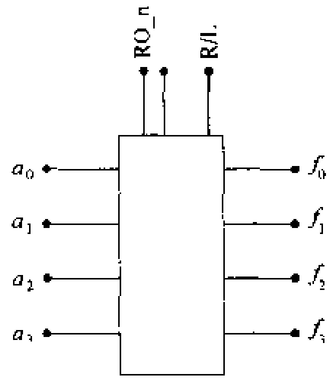


图 11.24 一般的循环器

$$f_3f_2f_1f_0 = a_2a_1a_0a_3 \tag{11.16}$$

而一位右循环得到

$$f_3f_2f_1f_0 = a_0a_3a_2a_1 \tag{11.17}$$

循环显示了转圈的行为，从一个字中移出去的一位被加到这个字的另一端。而移位操作是使一个 0 进入清空的位置上。如果修改这个单元以实现一位的左移操作，那么一个 $a_3\,a_2\,a_1\,a_0$ 的输入将产生如下的输出：

$$f_3f_2f_1f_0 = a_2a_1a_00 \tag{11.18}$$

这与右移操作的行为类似。

Verilog 提供如下按位移位的操作：

```
<<  //这是左移操作
>>  //这是右移操作
```

它可以用来说明向量移位；这两种移位都填充以“0”。这些操作的代码例子如下：

```
reg[7:0]a;
reg[7:0]new_1;
reg[3:0]new_2;
reg[3:0]b;
  new_1=a>>b;       //这把一个7位的字a右移b位
```

```
    new_2=a<<b;      //这把字a左移b位
    ...
```

一个循环操作可用许多不同的方式来说明。最简单的方式是一位一位地赋值，如在下面所列出的时钟单元：

```
...
reg[3:0];
always @(posedge clk)
  begin // 这是一位一位地左循环
    a[0]<=a[3];
    a[1]<=a[0];
    a[2]<=a[1];
    a[3]  <=a[2];
  end
...
```

通过增加控制位可以包含更为一般的循环操作。

在 VLSI 电路中可有许多不同的方式来实现循环和移位。虽然可以采用标准的以触发器为基础的设计，但基于 FET 开关特性的较为简单的电路可以实现高度规则的设计。考虑一个 4 位字的循环操作，图 11.25 的开关电路采用 4 个控制位 Ror_0，Ror_1，Ror_2，Ror_3 来说明一个 n 协作位的循环。这些信号是用组合逻辑产生的；在任何给定时刻，它们中间只有一个是 1。输入位 $a_3\ a_2\ a_1\ a_0$ 的信号循环是采用作为开关的 nFET 来实现的，这些开关按要求连接到输出线 $f_3\ f_2\ f_1\ f_0$。通过对 FET 重新连线，实现左循环阵列如图 11.26 所示。只要检查每列控制 FET 的输入-输出路径就可以验证这两个电路的连线是否正确。这两个阵列可以合成一个阵列，用控制字 Ro_n 来说明循环位数，而用另一个控制位来说明左/右循环。

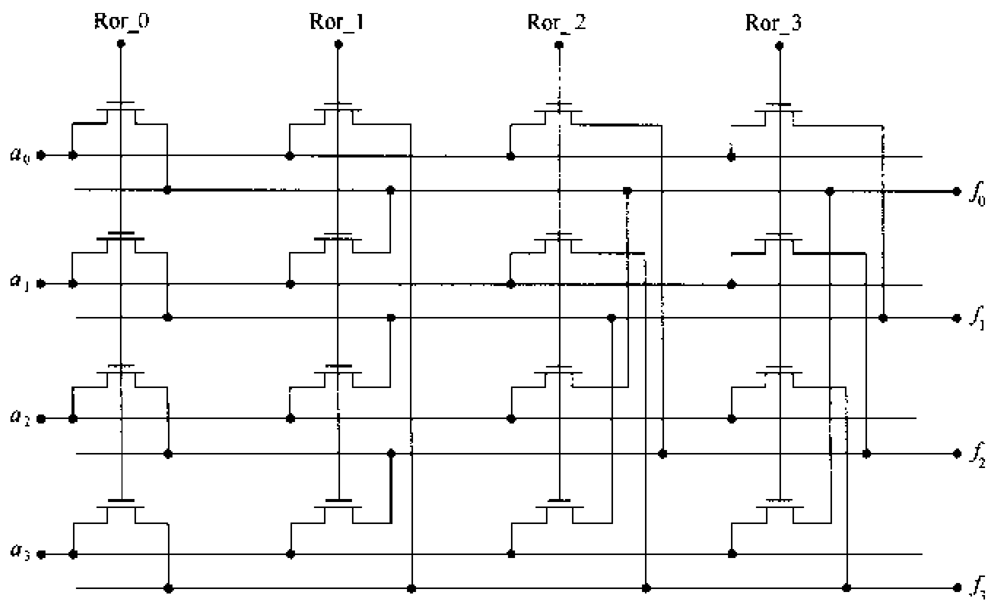


图 11.25　4 位右循环电路

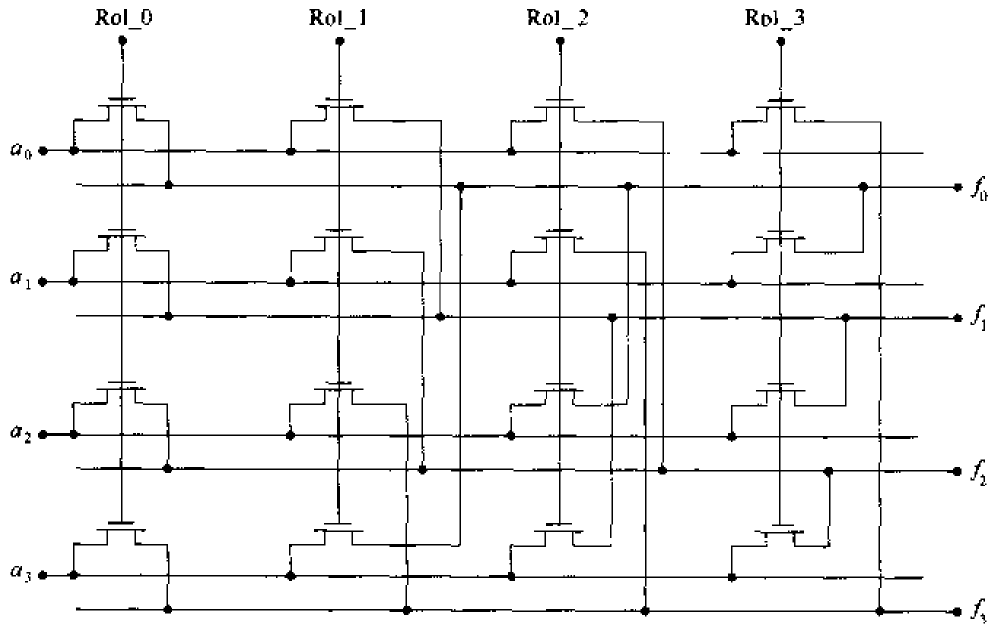
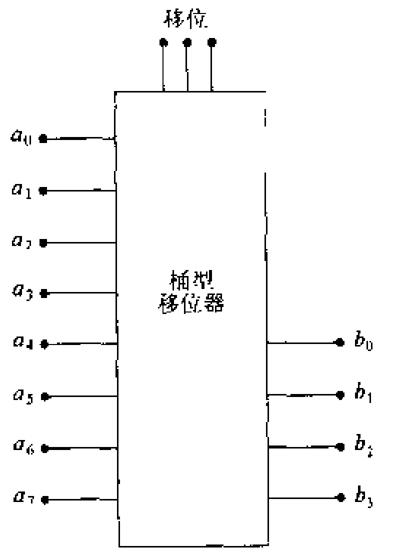


图 11.26 左循环开关阵列

一个与此相关的部件是**桶型移位器**，它是一个 $m \times n$ 阵列结构，其中 m 是输入的位数，而 n 是输出的位数。通常的情形是 $m = 2n$ 和 $m = n$。图 11.27 为一个 8×4 单元。控制字 shift(移位)的十进制值确定了用输入字 $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$ 表示的输出 $f_3\ f_2\ f_1\ f_0$，如图中表格概括的那样。这个单元可以用 nFET 阵列来构成，其结果如图 11.28 所示。与循环单元设计一样，每列管子由一位的信号 Sh _ n 来控制，在本设计中 n=0,1,2,3,4。这些控制信号是由一个组合逻辑电路从 3 位的控制字 shift 产生的。列控制信号中每次只有一个为 1，所以输入－输出路径是由有效的一列管子确定的。一个可以转圈的 4×4 电路就是一个循环单元。桶型移位器在 ALU(算术逻辑单元)中进行位操作是很有用的。这一电路的整体结构本身可以作为设计通信和并行处理用的集成交叉开关(cross-bar)网络和信道机(routers)的基础。



移位	$b_0\ b_1\ b_2\ b_3$
0	$a_0\ a_1\ a_2\ a_3$
1	$a_1\ a_2\ a_3\ a_4$
2	$a_2\ a_3\ a_4\ a_5$
3	$a_3\ a_4\ a_5\ a_6$
4	$a_4\ a_5\ a_6\ a_7$

图 11.27 8×4 桶型移位器

由于 nFET 阵列极为规则，这使它在 CMOS 物理设计层很容易画成版图。库单元可以像单个 FET 那样简单，也可以像构成较大阵列的 $p \times q$ 子单元那么复杂。只用 nFET 设计的主要缺点是阈值电压降问题（以及相应的弱 - 1 信号传送）和寄生参数对开关时间的限制。可以增加驱动器来加速电路和恢复电压摆幅。另一个方法是用传输门——替换 FET。虽然 pFET 消耗的面积很小，但布线的复杂性却会有较大程度的增加。

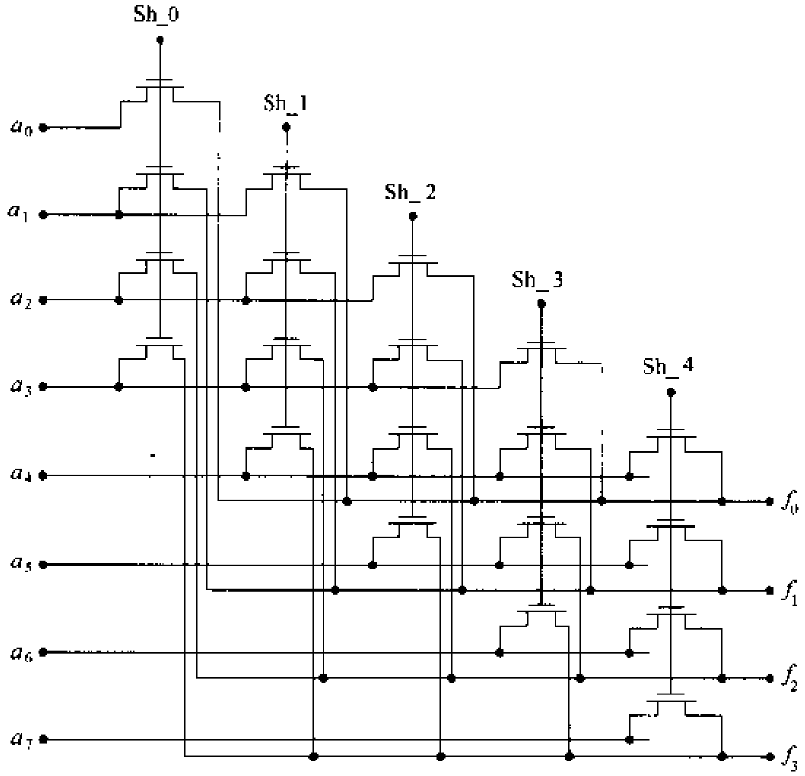


图 11.28 FET 阵列桶型移位器

11.6 锁存器

锁存器(Latch)是能接收和维持一个输人二进制位的部件。简单的 D 锁存器是许多设计的基础。图 11.29(a)为 D 锁存器的符号，而图 11.29(b) 则为它的逻辑图。通过观察可以看到，这个电路是由一个以 NOR 门为基础的 SR 锁存器及两个互补的输人构成的。锁存器是透明的，即 D 的变化在经过了电路的延迟时间后可以在输出端 Q 和 $\overline{Q}$ 上看到。

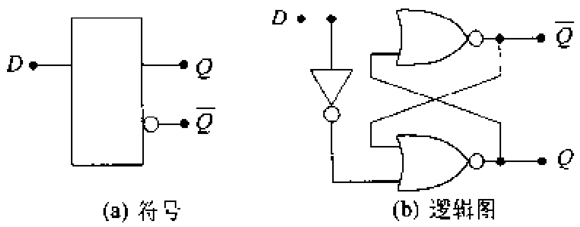


图 11.29 D 锁存器

这一部件的行为描述为：

```
module d_latch(q,q_bar,d);
input d;
output q,q_bar;
reg q,q_bar;
always @(d)
```

```
    begin
      #(t _ d)q = d;
      #(t _ d)q _ bar = ~ d;
    end
  endmodule
```

采用 t _ d 表示时间延迟。这一陈述模拟了能维持输入状态的交叉耦合 NOR 电路的作用。等效的结构描述为：

```
module d _ latch _ gates(q,q _ bar,d);
input d;
output q,q _ bar;
wire not _ d;
not(not _ d,d);
nor #(t _ nor)g1(q _ bar,q,d),
    #(t _ nor)g2(q,q _ bar,not _ d);
endmodule
```

它在电路和物理设计层次上为该部件提供按逻辑门设计的方法。

1. **组合逻辑设计**(Combinational Logic Design)

CMOS 电路可用逻辑图或结构描述来设计，图 11.30 为一个直接的转换。在物理层次上，它可通过引用两个 NOR2 单元和一个 NOR 单元，再加上互连线来构成。另一种方法则是采用全定制版图，它也许占用较少的面积。

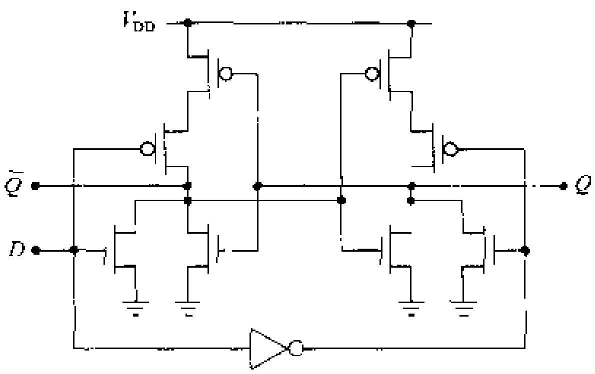
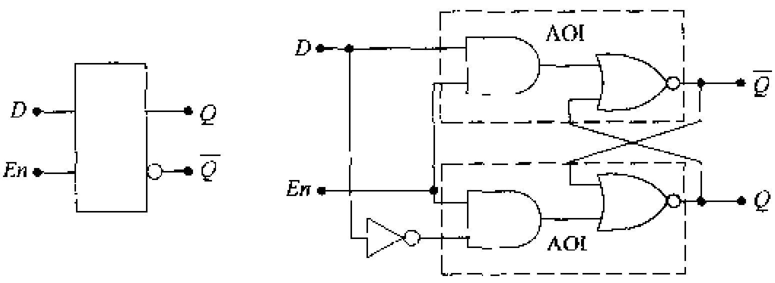


图 11.30　CMOS D 锁存器电路

可把一个使能控制 En 加入基本的 D 锁存器中，即把这个输入连到两个 AND 门如图 11.31 所示。使能信号 $En=0$ 将封锁 AND 门的输入，因为它迫使 AND 门的输出为 0，于是使 SR 锁存器进入维持状态。如果 $En=1$，则 D 和 $\overline{D}$ 的值被允许进入 NOR 电路。为了把这一控制包括在行为描述中，重写代码如下：



(a) 符号　(b) 逻辑图

图 11.31　具有 *Enable* 控制的门控 D 锁存器

```
module d _ latch(q,q _ bar,enable);
input d,enable;
```

```
output q,q _ bar;
reg q,q _ bar;
always @(d and enable)
  begin
    #(t _ d)q=d;
    #(t _ d)q _ bar= ~d;
  end
endmodule
```

为了改变状态,要求 enable=1。

结构描述的基础是不厌其烦地列出电路。由于 CMOS 允许把复杂的逻辑门作为基本单元,所以先描述复杂门然后在写代码时再例举它就更为合理。

```
// 首先定义 AOI 模块
module aoi _ 2 _ 1(out,a,b,c);
input a,b,c;
outout out;
wire w1;
and(w1,a,b);
nor(out,w1,c);
endmodule
// 现在用它来构成锁存器
module d _ latch _ aoi(q,q _ bar,enable);
input d,enable;
output q,q _ bar;
wire d _ bar;
not(d _ bar,d);
aoi _ 2 _ 1(q _ bar,d,enable,q);
aoi _ 2 _ 1(q,enable,d _ bar,d,q _ bar);
endmodule
```

名字为 aoi _ 2 _ 1 的模块是一个 AOI 门,其中 AND 有两个输入,OR 有一个输入。尽管这一表示法并不标准,但它却广泛地运用在实践中。注意当例举这一模块时,输入的顺序必须保持一致。D 锁存器的 CMOS 电路实现显示在图 11.32 中,它包括一个反相器和两个 AOI 电路。

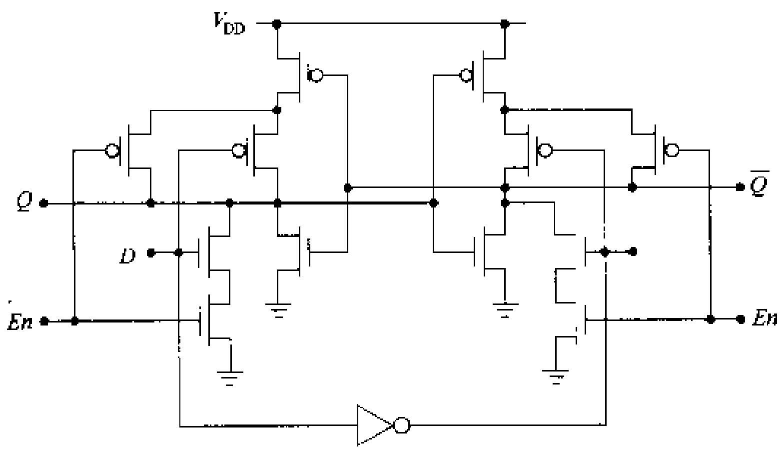
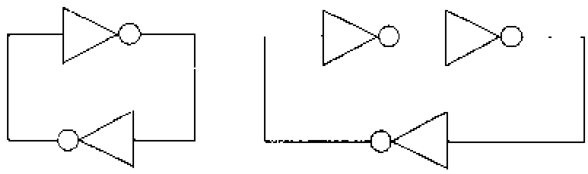


图 11.32 AOI CMOS 门实现带 *Enable* 信号的 D 锁存器

2. CMOS VLSI 锁存器(CMOS VLSI Latch)

许多静态 D 锁存器的 CMOS VLSI 电路是用反相器及 TG 或 FET 传输管来构成的。这一设计是以称为**双稳电路**的最简单静态存储方式为基础的。一个双稳电路能够永久地(或者至少说只要在电源存在的情况下)存储(或维持)一个逻辑 0 或一个逻辑 1。

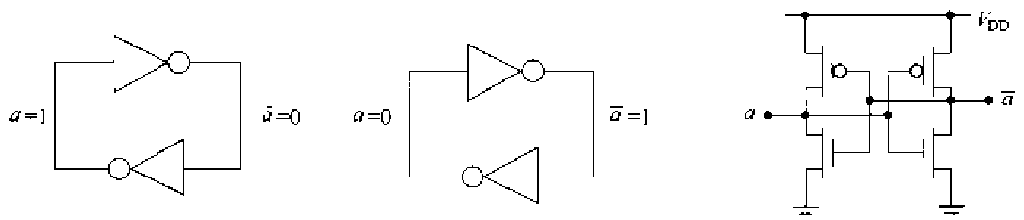
一个基本的双稳电路包含两个反相器如图 11.33(a)所示。这两个反相器连接成一个反相器的输出为另一个的输入,从而形成一个闭环。任何一个具有偶数个反相器的闭环都可以构成一个双稳电路。如果运用三个反相器如图 11.33(b)所示,那么所形成的电路就是不稳定的,不能保持一个二进制位的值。一个具有奇数个反相器的闭环常常称为**环振荡器**,因为在任何一点的信号随时间振荡。



(a) 双稳　　(b) 环振荡器

图 11.33　闭环反相器组合

双稳电路的存储原理如图 11.34(a)所示。如果左边的值 $a=1$,那么跟踪通过上面一个反相器的信号路径表明右边的值为 $\bar{a}=0$。如果通过下面的反相器继续跟踪信号到左边,将得到 $a=1$ 的起始点。这表明状态 $a=1$ 是稳定的,因为它能由电路本身来维持。同样的理由也可以应用到 $a=0$ 的情形,如在同一图中所表示的那样。图 11.34(b)的 CMOS 电路运用两个反相器实现双稳电路。



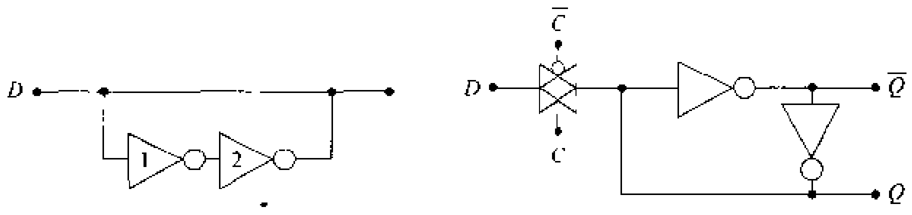
(a) 稳定状态　　(b) CMOS 电路

图 11.34　双稳电路工作情况

为了构成一个 D 锁存器,必须为输出信号提供一个入口节点。一个简单的设想就是图 11.35(a)所示的**接收电路**。D 的值由反相器对构成的双稳电路保持。双稳电路帮助图中的这条直线阻止 D 的变化,使这条直线可以作为外部信号线驱动的接收器模块的输入级。从反相器 2 的输出至反相器 1 的输入的**反馈环**提供希望的锁存功能,却使设计变得复杂。反相器 1 需要检测输入的变化,但反相器 2 又不能太强,以免阻止状态的变化。一般地反相器 1 可以采用较大的 FET,但反相器 2 则有意采用小的管子而使之较弱。

在输入端增加一个传输门如图 11.35(b)所示,可以提供控制装载的功能。当 $C=0$(所以 $\bar{C}=1$)时,TG 的作用如一开路的开关,于是该电路的输出端保持 Q 和 $\bar{Q}$ 的值。当控制位被置成 $C=1$($\bar{C}=0$)时,TG 导通于是允许输入 D 传送到锁存电路中。在这期间,锁存器是透明的,输出变为 $Q=D$ 及 $\bar{Q}=\bar{D}$。如果 C 被置 0,则该状态保持不变。因此控制位 C 相当于前面曾采用的使能信号 En。对反相器设计的约束仍然成立。

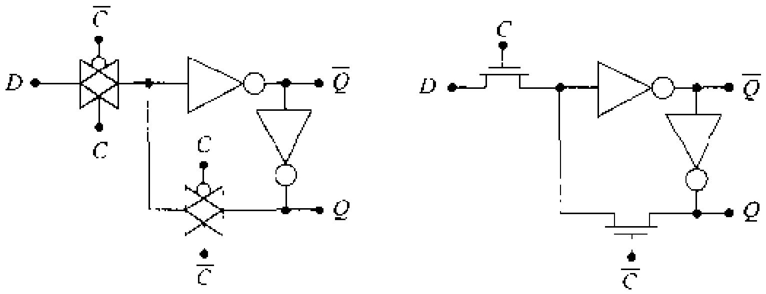
虽然这些电路很容易用 CMOS 实现,它们却是比较慢的部件。这是因为双稳态电路力图保持所存放的值而阻止变化,如对图 11.34(a)所讨论的那样。如果迫使所存储的电压发生变



(a) 接收电路　　(b) 控制装载

图 11.35　在双稳电路上加入输入节点

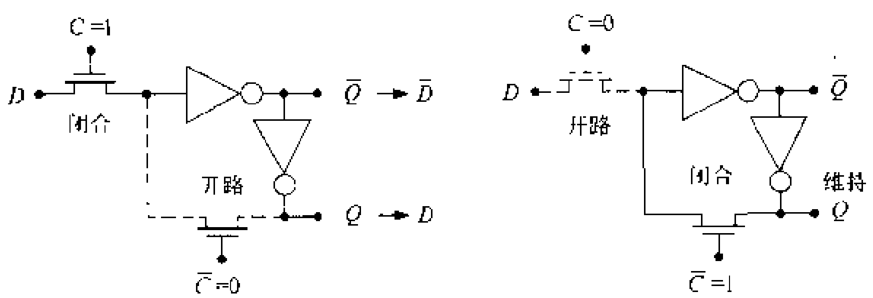
化，那么所存在的从输出回到输入的反馈就会阻止这一过渡。解决这个问题的方法是增加另一个开关来切断数值得以存储的反馈环。图 11.36(a)所示的 TG 电路通过在两个反相器间增加一个控制相位相反的开关来达到这一目的。在许多情形中，芯片的设计者乐于采用 nFET 来代替 TG 以使布线较为简单，这显示在图 11.36(b)中。注意输入 FET 由 C 控制，而反馈回路中的晶体管则由反相的 $\overline{C}$ 来切换。



(a) CMOS TG 方式　　(b) nFET 传输门方式

图 11.36　用相位相反的一对开关实现 D 锁存器

nFET 锁存器的操作总结在图 11.37 中。当 $C=1$ 时输入数据 D 被装入，如图 11.37(a)所示，这使输入 FET 导通并使反馈环路切断。输入端因此见到一个简单的 NOT 链，能够非常快地装载。控制位 $C=0$ 确定了维持状态，如图 11.37(b)所示。这使输入 FET 截止但建立起反馈回路，从而保持住 Q 和 $\overline{Q}$ 的值。由于 TG 的逻辑方程与 nFET 的完全一样，所以这一描述也适用于以 TG 为基础的电路。



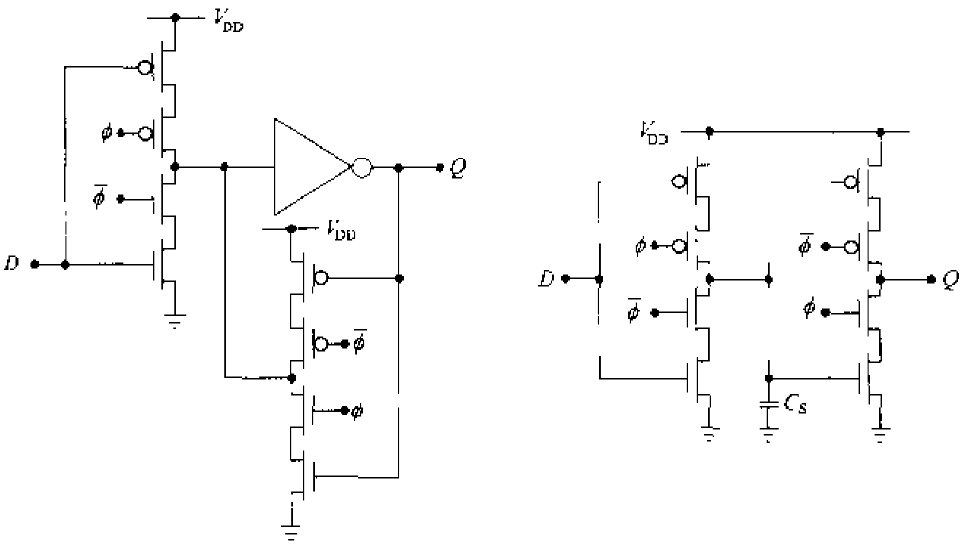
(a) $C=1$ 时装载　　(b) $C=0$ 时维持

图 11.37　D 锁存器的操作

C^2MOS 电路提供另一种 D 锁存器设计类型的基础。如图 11.38(a)所示的锁存器采用 C^2MOS 反相器作为输入级且由时钟控制装载。当 $\phi=0$ 时，D 被允许进入该电路并通过静态反

相器，于是在经过了一个特征延迟时间之后 $Q=D$。当时钟过渡到 $\phi=1$ 时，第一级被驱动至 Hi-Z 状态（高阻状态），而反馈回路则被控制相位相反的反相器闭合，于是维持输出直到下一个时钟周期。

图 11.38(b)为另一种运用一对串联 C^2MOS 反相器来构成锁存器的方法。这个电路的操作有相当的不同，即它是一个真正的动态电路，也就是它运用了电荷在电容节点上的存储效应。在前后两级之间寄生电容 C_s 的作用是作为该电路的存储部件。当 $\phi=0$ 时，D 被允许进入，于是相应于 $\overline{D}$ 的电荷被存放在 C_s 上。当时钟变为 $\phi=1$ 时，第一级的输出处在 Hi-Z(高阻)状态，它维持输出在 C_s 上的电荷。这一时钟相位使输出为 Q 的第二级工作，因此 Q 的值将是 $\overline{D}$ 的值延迟一段该电路的上升或下降时间。应当注意电荷泄漏限制了 C_s 维持该状态的时间。尽管该电路在这里是作为锁存器介绍的，但它常用来作为同步电路中的时延单元。



(a) C^2MOS 静态锁存器　　(b) 动态锁存器

图 11.38　基于 C^2MOS 的 D 锁存器电路

11.7　D 触发器

触发器与锁存器的不同之处在于它不是透明的。D 触发器(D Flip - Flop，DFF)是在 CMOS 电路中最常用的触发器。基本的 DFF 设计是一个主从形式的触发器，它是把两个控制相位相反的 D 锁存器串联在一起得到的，如图 11.39 所示。时钟信号 ϕ 控制操作并提供同步信号。主锁存器允许 D 在 $\phi=0$ 时输入，M1 的行为如同一个闭合的开关。在此期间 nFET M2 和 M3 开路。当时钟过渡到 $\phi\to 1$ 时，开关 M2 和 M3 闭合使这一位传送到从触发器。此时主触发器的输入被封锁，因为当 $\phi=1$ 时 M1 开路。这一主从电路的行为如同一个**正边沿触发**的器件，因为在正时钟边沿处的 D 值确定了传送到“从触发器”并在 Q 端出现的值。[②] 再次注意到，一个二进制位在时间 t_1 时被锁存入“从触发器”，但它直到

$$t_1+t_{FET}+t_{NOT} \tag{11.19}$$

② 就最严格的意义上讲，一个主从触发器与一个真正的正沿触发电路是有区别的。然而在 VLSI 设计中，这一术语常互用。

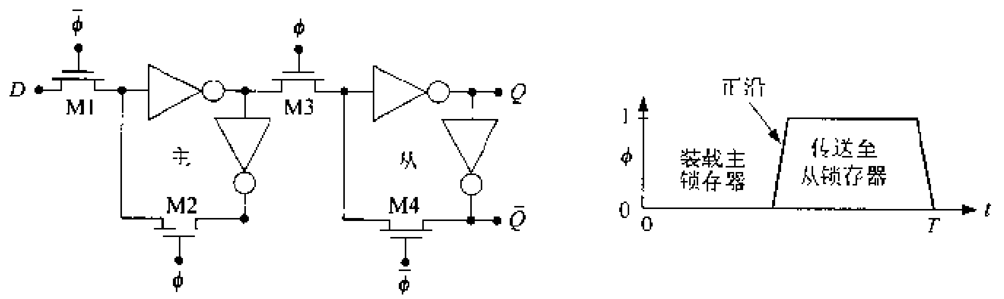


图 11.39 主从 D 触发器

之后才出现在输出端。式中 t_{label}是 label 所说明的元件的上升和下降时间。图 11.40(a)为正沿触发 DFF 的符号。"小三角"表示边沿触发的输入端。在图 11.40(b)中增加一个小圆圈就构成了一个**负边沿触发** DFF 的符号。在电路一级,全部所需要的只是互换 ϕ 和 $\bar{\phi}$ 信号。

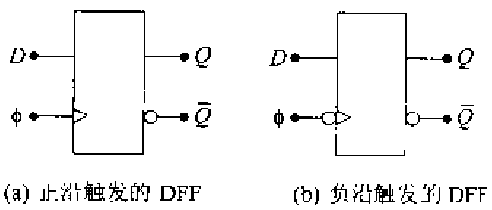


图 11.40 边沿触发 DFF 的符号

一个正沿触发 DFF 的 Verilog 行为描述可以写成如下:

```
module positive_dff(q,q_bar,d,clk);
input d,clk;
output q,q_bar;
reg q,q_bar;
always @(posedge clk)
  begin
    q=d;
    q_bar=~d;
  end
endmodule
```

在实际应用中,还需要有一组延迟时间的说明。对 **always** 语句做如下修改就可以得到一个负边沿触发的模块:

```
always @(negedge clk)
```

它改变激励的方式是很明显的。用 **nmos** 和 **not** 两个基本元件模拟图 11.39 电路的结构描述是很容易的,因此留给读者作为练习。

这里要重复的一个重要观点是 CMOS 电路的设计决定了通过 DFF 的延时。考虑图 11.41 所示的另一个电路,它在逻辑上等同于图 11.39 所画的电路,但从输入 D 到输出 Q 的数据通路经过了四个反相器而不是两个。由于每个逻辑门都会引入附加的信号延时,因而这一电路将比原先的设计慢。因此可以看到电路的拓扑连接和由此产生的物理设计直接影响 HDL 代码描述的高层次结构的速度。这一类考虑是区分高速 VLSI 与其他数字系统设计的因素之一。

通过改变门的功能有可能把直接"清零"(clear)和"置位"(set)功能加入这个电路中。一种方法就是运用 NAND2 逻辑。考虑图 11.42 的情形。这里一个输入是控制位 s 而另一个则是数据 in。当 $s=0$ 时,输出为 1 而不管 in 是什么值。如果 $s=1$,则 $out=\overline{in}$,如图所示。用 NAND2 门替代所选的反相器可以得到具有低电平清零、置位或同时具有清零和置位的 DFF。

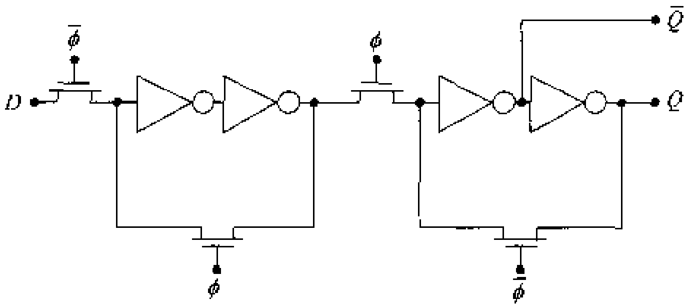
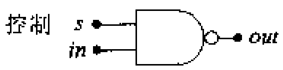


图 11.41　另一种主从 DFF 电路

图 11.43(a)提供了用 Clear 控制清除锁存器内容(至零)的功能。当 Clear = 1 时,NAND 门的作用如同反相器,因此电路的行为如同一个通常的 DFF。当 Clear = 0 时,在"从锁存器"中的 NAND 迫使输出 $Q=0$。主触发器的输出被迫使为逻辑 1,它经反相使输出 $Q=0$。描述这一部件的 Verilog 代码如下:



s	in	out
0	0	1
0	1	1
1	0	$\overline{in}$
1	1	$\overline{in}$

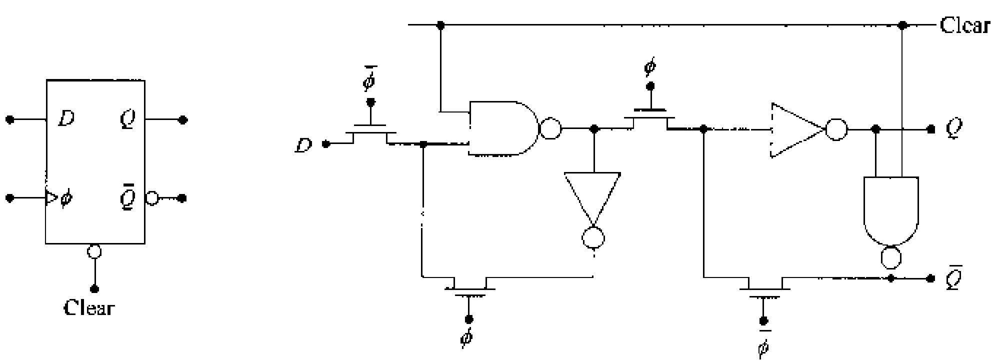
图 11.42　NAND2 用作控制元件

```
module dff_clear(q,d,clear clk);
input d,clk,clear;
output q;
reg q;
always @(posedge clk)
     q=d;
always @(clear)
   if(clear)
     assign q=0;
   else
     deassign q;
endmodule
```

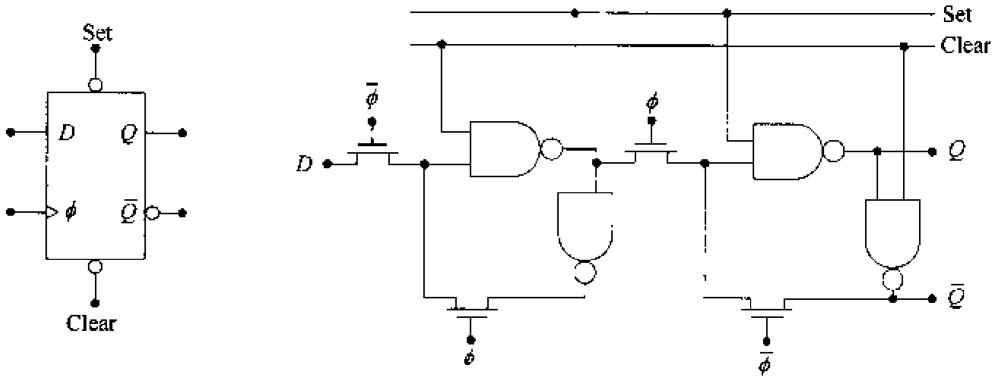
以上代码运用了 **deassign** 语句,它在 clear 为 0 时执行,这使 q 的值返回到它在 q = d 这一行代码中建立起来的值。

同时具有清零和置位控制的 DFF 如图 11.43(b)所示。置位功能是通过用 NAND2 门替代其余两个反相器而实现的。Set = 1 为通常的操作,Set = 0 迫使"从触发器"的输出 $Q=1$。而这又被"主触发器"的输出加强。注意 Clear 和 Set 不能在同一时间都为 0,否则这会迫使 $Q=\bar{Q}$。通常在单元库中可以找到几种不同的 DFF,它们具有的特点如输入缓冲器,时钟缓冲器以及具有组合逻辑门的输入。例如一个 T 触发器(TFF)在每个时钟上升沿处改变它的状态,它可以通过增加从 $\bar{Q}$ 至 D 的反馈回路来实现,如图 11.44 所示。低电平置位逻辑使这个触发器的初始值设定为 1。

一个基本的 DFF 在每个时钟边沿装载一个新的数据。为了存储任何数目的时钟周期,可



(a) 具有直接清零的 DFF



(b) 具有直接清零和置位的 DFF

图 11.43　具有低电平清零及清零/置位控制的 DFF 电路

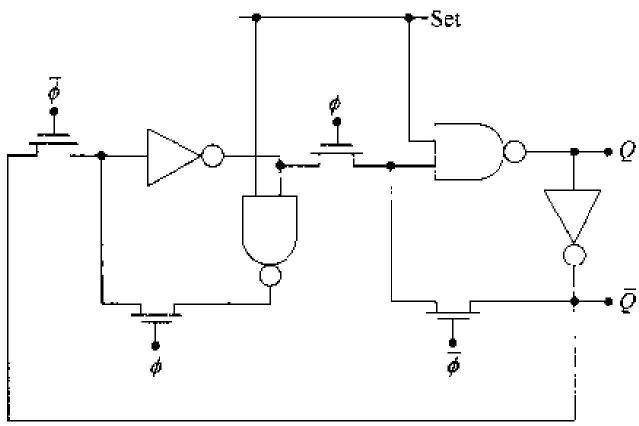
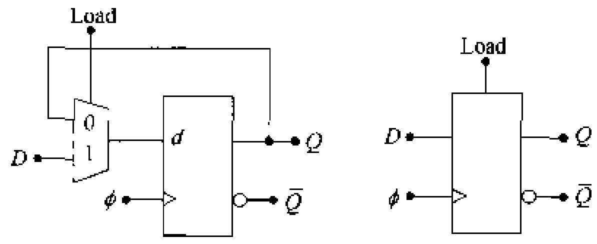


图 11.44　利用反馈将 DFF 改为 TFF

在该电路上增加一个控制信号及相关的逻辑。一个简单的经典解决办法显示在图 11.45(a)。控制信号 Load 控制一个 2:1 的 MUX。如果控制信号有效即 Load = 1,则 MUX 允许一个新的数据 D 进入到 DFF。当控制位的值 Load = 0 时,则取出输出 Q 并且重新把它放回输入端。这类电路通常集成为单个元件,它简化的符号如图 11.45(b)。包括 Load 控制的简单的 Verilog 描述如下:

```
module dff_load(q,q_bar,d,load,clk);
input d,clk,load;
```

```
output q,q_bar;
reg q,q_bar;
always @(posedge clk)
  begin
    if(load)q=d;
    q_bar=~d;
  end
endmodule
```



(a) 连线图　　　　(b) 符号

图 11.45 具有 Load(装载)控制的 D 触发器

设计具有控制装载 DFF 的另一种方法显示如图 11.46 所示。这是一个基本的 CMOS 主从触发器,但把控制信号改为

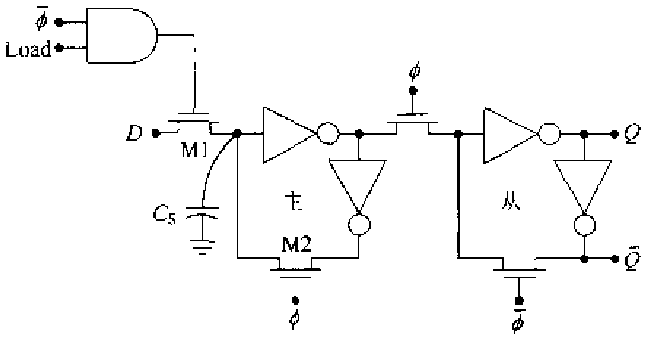


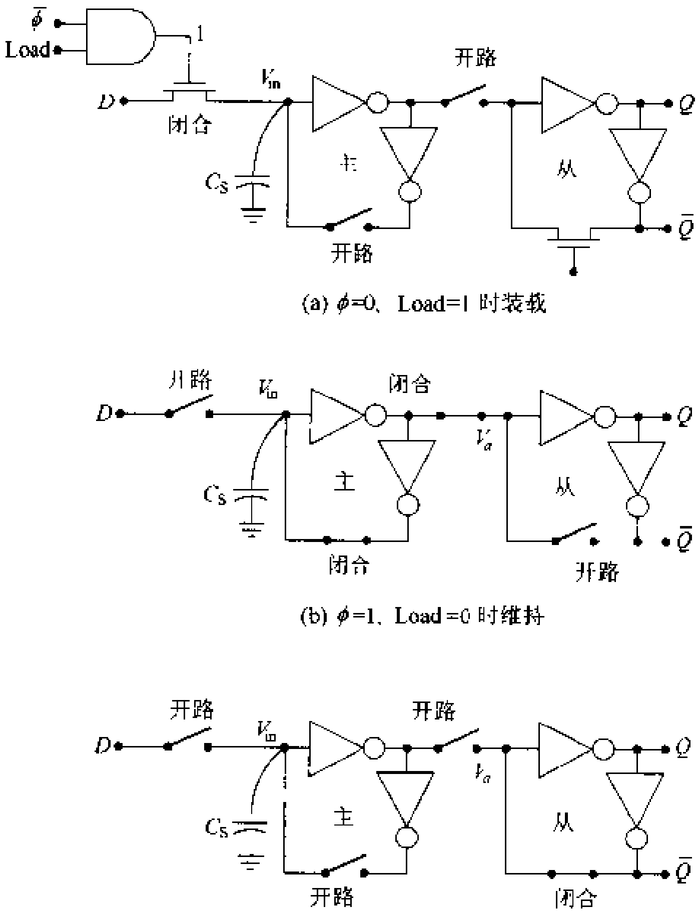
图 11.46 带有 Load(装载)控制的 CMOS 主从触发器

$$\bar{\phi}\cdot \text{Load} \tag{11.20}$$

并送到输入 FET M1 上。只有当 Load 和 $\bar{\phi}$ 同时为 1 时才允许装载。该电路的操作显示在图 11.47 中。图 11.47(a)为装载的情形。它与原来电路的操作完全相同。在主触发器电路的输入端处,D 的值使电压 V_{in}为 0V 或 V_{max},这分别相应于逻辑 0 或逻辑 1。注意这一电压建立了存储电容 C_s 上的电荷状态,如电路图中所示。

当 Load=0 时,数据保持状态有两种可能的情形。当 $\phi=1$ 时,主锁存器依靠闭环反馈回路保持数据值,并且把它传送给"从锁存器"和输出端,如图 11.47(b)所示。另一种情形是如果 $\phi=0$ 和 Load=0 同时发生,那么电路开关处在如图 11.47(c)所示的状态。主触发器的反馈回路开路。存储功能是通过保持 C_s 上的电荷来实现的,这使它成为一个受电荷泄漏影响的准动态电路。注意此时"从触发器"的反馈回路是闭合的。于是就建立起电压 Va。当时钟返回到 $\phi=0$ 时,又变成如图 11.47(b)所示的电路。这会有两方面的影响。首先,主触发器的反馈回路闭合从而建立起静态保持功能。其次,从触发器电压 V_a 加强了在 C_s 上的电压值。应当记住电荷泄漏会限制可用时钟频率的下限。特别是当时钟停止后,在时钟再次启动时会

使电路出现一个相当长的延时。这在开发测试这一设计电路的方法时可能会有明显的差别。



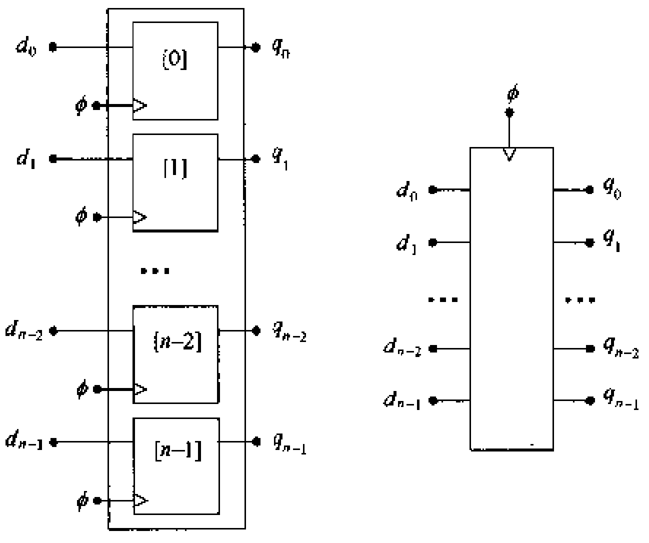
(a) ϕ=0、Load=1 时装载

(b) ϕ=1、Load=0 时维持

(c) ϕ=0、Load=0 时维持

图 11.47 具有装载控制的 CMOS DFF 的操作

11.8 寄存器



(a) 内部结构　　(b) 基本符号

图 11.48 构成一个 n 位的寄存器

寄存器是能把一个字作为一个整体存放起来的一类电路的统称。一位的寄存器就是一个触发器，而一个 n 位的寄存器则装载和保持一个 n 位的字。寄存器是 VLSI 设计中的重要部件。它能设计构成现代数字系统基础的时序电路和状态机。从单个位的逻辑转为处理字的单元是在层次化设计中关键的一步。

一个 n 位正沿触发寄存器可以通过并行组合 n 个单个的 DFF 来构成，如图 11.48(a)所示。图 11.48(b)所示的寄存器符号避开了各个电路的细节但却提供基本的操作：在时钟上升沿时 n 位的输入字装

载到寄存器中。输出则在电路延迟时间后有效,正如在前一节中讨论的那样,可以增加各种保持功能。

由于 DFF 在本质上是时钟控制的,因此它们会影响电路的动态功耗。前面在图 11.47 中描述的 DFF 操作显示了动态功耗的两个来源。一个是由于控制主从触发器的工作所需要的 4 个时钟 FET。这 4 个时钟 FET 是时钟驱动器的负载,它们增加了动态功耗

$$P_{dyn} = C_L V_{DD}^2 f \tag{11.21}$$

相当于加大负载电容 C_L。这发生在每个周期。另一个功耗的来源是对存储节点电容 C_s 的再充电以补偿电荷泄漏,这相对较小。如果输入值改变,那么反相器也消耗功耗,但这是不可避免的;逻辑操作总是要求能量的。

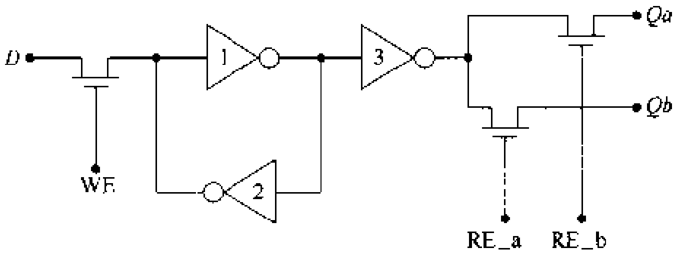


图 11.49　一位静态多输出口寄存器电路

一个纯静态多输出端口的电路如图 11.49 所示。这个电路是一个基本的两个反相器存储电路加上读写管以及一个输出驱动器。写使能信号 WE 控制装载,而两个读使能信号(RE_a 和 RE_b)用来使输出分配到 Qa 或 Qb。由于这个电路运用单端输入,所以反相器的设计十分重要。反相器 1 应当对输入十分灵敏,并且具有足够的驱动强度以能迅速响应。因此这个反相器和反相器 3 采用较大的管子是比较合适的。反相器 2 提供反馈以便锁存数据。由于它会阻止在输入端的变化,所以它通常采用小管子设计得非常弱。这些考虑与在讨论图 11.35(a)中的锁存器时是完全一致的。时钟并没有包括在这个基本电路中,如果需要则应增加到输入端处。一个简单的解决方法是增加一个时钟控制的 nFET 与输入晶体管串联,或对现有的这个电路建立一个组合的控制信号 WE·ϕ。

图 11.50 为用这个电路构成的一个 n 位寄存器。它的每位提供一个输入和两个输出。如果需要可再增加另外的端口。由于这个设计没有相应的时钟功耗,所以它在寄存器内容需要保持一段长时间的场合十分有用。特别是它可以用来构成**寄存器堆**,后者实际上是存储字的寄存器的集合。

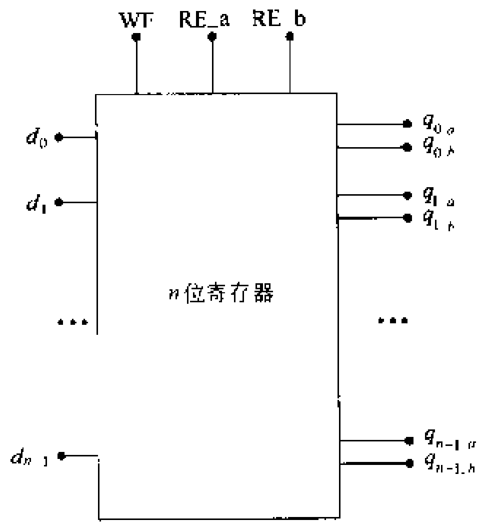


图 11.50　n 位静态多输出口寄存器

11.9　综合的作用

本章中的逻辑部件用来说明把一个高层次的 HDL 描述向下转变成逻辑或电路图的技术。

在每个情形中，较大规模的功能可以采用不同的逻辑电路或CMOS设计类型来实现。由于一个给定功能的每种实现将产生不同的开关速度和占有不同的硅片面积，因此设计者的作用会由于需要做出正确的选择而变得复杂。CAD工具允许进行不同的设计并且估计它们的特性以便比较。这对于设计关键的数据通路或复杂的逻辑电路特别重要。

体系结构的细节处于层次化设计的顶层。高层次的抽象描述用来定义和验证系统的工作。一旦在这个层次上的模型是合理的，则向下进行到描述层次以用来作为逻辑综合的基础。逻辑综合工具设计把HDL代码转变成逻辑电路，这些电路是用预先定义的存放在库中的基本单元来构成的。只要基本逻辑电路与CMOS单元之间保持一一对应关系，逻辑工具的输出就可以用来构建电路。

综合是设计自动化最重要的特点之一。它把许多很枯燥的工作交给机器来完成。然而就面积或速度而言，综合的结果并不总是“最好”的。在关键单元，也许有必要对逻辑、电路或两者都进行全定制设计以达到所需要的特性。本章的例子说明了这一过程的复杂性，以及在各个不同层次间的相互作用。

11.10 参考资料

[1] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[2] Kerry Bernstein, et al., **High-Speed CMOS Design Styles**, Kluwer Academic Publishers, Norwell, MA, 1998.

[3] Michael D. Ciletti, **Modeling, Synthesis, and Rapid Prototyping with Verilog HDL**, Prentice Hall, Upper Saddle River, NJ, 1999.

[4] Randy H. Katz, **Contemporary Logic Design**, Benjamin/Cummings, Redwood City, CA, 1994.

[5] Pran Kurup and Taher Abasi, **Logic Synthesis Using Synopsys®**, 2nd ed., Kluwer Academic Publishers, Norwell, MA, 1997.

[6] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2001.

[7] Douglas J. Smith, **HDL Chip Design**, Doone Publications, Madison, AL, 1996.

[8] Michael J.S. Smith, **Application-Specific Integrated Circuits**, Addison-Wesley Longman, Reading, MA, 1997.

[9] John P. Uyemura, **A First Course in Digital Systems Design**, Brooks/Cole Publishers, Monterey, CA, 2000.

[10] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[11] Neil H. E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, Reading, MA, 1993.

11.11 习题

[**11.1**] 考虑构成一个 4:1 的 MUX 的问题。

(a) 采用传输门设计一个 4:1 的 MUX。

(b) 用 Verilog cmos 基本单元写出你的电路的结构描述。

(c) 画出 MUX 的 RC 等效电路。假设 TG 的电阻为 R,这个开关每边的电容为 C。然后利用 Elmore 公式求出通过这个 MUX 的最坏情形路径的延迟时间常数。

[**11.2**] 基于 assign 语句写出一个 16:1 MUX 的 Verilog 模块。采用一个 4 位的选择字 $s_3 s_2 s_1 s_0$ 把所选择的输入 $p_i(i=0,\ldots,15)$映射到输出端。

[**11.3**] 考虑用较小的 MUX 作为基本单元构成一个 8:1 MUX。

(a) 采用 4:1 和 2:1 MUX 单元来构成一个 8:1 MUX。

(b) 选择一种逻辑电路来实现这个设计。

(c) 假设逻辑门由静态 CMOS 电路构成。如这个 8:1 MUX 驱动一个等于 $10C_{inv}$的电容 C_{out}(这里 C_{inv}是单位反相器的电容),试用逻辑努力(Logical Effort)的方法来设计这些门。

[**11.4**] 在主要逻辑路径中仅采用传输门来设计一个 2/4 高电平有效的译码器。然后采用 cmos 基本单元写出你的电路的 Verilog 描述。

[**11.5**] 用 NOR 门设计一个 2/4 低电平有效的译码器。

(a) 列出 Verilog 结构描述。

(b) 修改你的 Verilog 代码以便包括一个输入使能控制,然后构成这个新电路。

[**11.6**] 采用标准的动态或多米诺 CMOS 逻辑设计一个 4:1 MUX。

[**11.7**] 设计一个 4 位左/右循环单元,其中循环位数用 2 位字 Ro _ n 说明,循环方向用一位 R/L 来说明,R/L=1 表示右循环,R/L=0 表示左循环。设计时以图 11.25 和图 11.26 的 FET 阵列为基础。

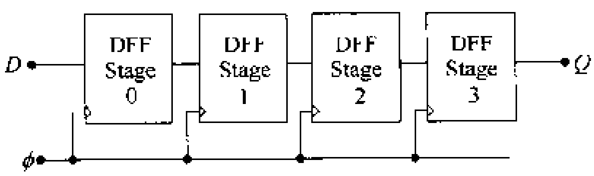


图 P11.1

[**11.8**] 写出图 11.14 所示的 8 位相等检测电路的 Verilog 结构描述。

[**11.9**] 考虑图 P11.1 中所示的 4 位移位寄存器。数据流 D 由顺序 d_0, d_1, d_2 和 d_3 的二进制位序列构成。时序的安排为:d_0 在第一个时钟沿进入第 0 级,在下一个上升沿 d_1 进入第 0 级而 d_0 移至第一级,依次类推。

(a) 采用 DFF 模块作为基本单元写出这个移位寄存器的 Verilog 描述。你可以采用本书中的 DFF 模块或写出你自己的模块。

(b) 对 DFF 选择一种 CMOS 设计技术且用它来构成这个电路。

(c) 采用 nmos 和 pmos 基本单元写出这个移位寄存器的 Verilog 描述。

[**11.10**] 建立起一个 8 位寄存器的 Verilog 模块。这个寄存器在且仅在控制位 En=1 时在时钟上升沿装载字。你可以采用任何层次的描述。

[**11.11**] 写出图 11.49 所示寄存器电路的结构描述。

第 12 章　CMOS VLSI 运算电路

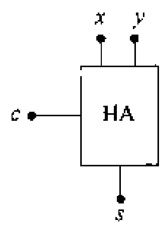
运算功能如加法和乘法在 VLSI 设计中具有特殊的意义。许多应用要求这些基本操作，但自早期数字电路芯片问世以来，好的芯片实现一直是一个挑战。本章将详细考察二进制加法器，且把讨论扩展到乘法器。

12.1　一位加法器电路

考虑两个二进制数字 x 和 y。它们的二进制和用 $x+y$ 表示如下：

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+0 &= 1 \\ 1+1 &= 10 \end{aligned} \tag{12.1}$$

这里最后一行的结果是二进制数 10(即是十进制中的 2)。这一简单的例子说明了有关加法的问题。如果取两个 r 进制的数字，其范围为 $0,1,\cdots(r-1)$，那么这两个数的和可能超出这一组数字本身的范围。自然这就是“进位输出”概念的来源。在“二进制和”1 + 1 中，其结果 10 被看成是一个 0 以及一个左移的 1，后者即为“进位输出 1”。



x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

图 12.1　半加法器的符号及操作

一个**半加法器**电路有两个输入(x 和 y)以及两个输出(和 s 和进位输出 c)，它可以用图 12.1 的表格来描述。输出由以下的基本方程给出：

$$\begin{aligned} s &= x \oplus y \\ c &= x \cdot y \end{aligned} \tag{12.2}$$

它们可以直接从表中推导出来。这个单元的 Verilog 高层次行为描述可以写成：

```
module half_adder(sum,c_out,x,y);
input x,y;
output sum,c_out;
assign {c_out,sum} = x + y;
endmodule
```

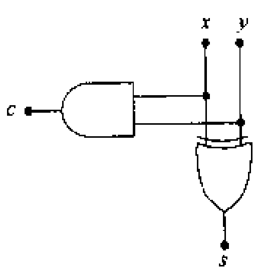


图 12.2　半加器的逻辑图

这里 x 和 y 定义为单个二进制位，然后运用链接算子{ }得到两位的结果。这一算子按列出的顺序“连接”两个二进制部分以形成单一的结果。另一方面，也可构成如图 12.2 所示的门级电路。这是由如下的结构模型描述的：

```
module half_adder_gate(sum,c_out,x,y);
input x,y;
output sum,c_out;
and(c_out,x,y);
xor(sum,x,y);
endmodule
```

它运用了基本门的例举。图 12.3 则为另外两种可能的电路。图 12.3(a)的半加器采用 NAND2 门;而图 12.3(b)的另一个电路则是基于 NOR 的设计。也许 NAND 设计比较合适,因为它避免了串联的 pFET 链,但是由于半加器非常简单,所以这个差别不是一个主要因素。

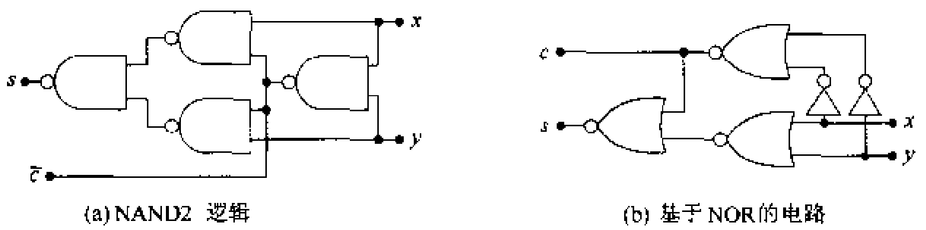


图 12.3 另外两种半加器的逻辑电路

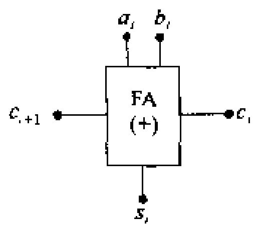
一个比较复杂的问题是 n 位相加的二进制字。考虑两个 4 位数 $a = a_3\,a_2\,a_1\,a_0$ 和 $b = b_3\,b_2\,b_1\,b_0$。把它们相加得到

$$\begin{array}{r} a_3a_2a_1a_0 \\ +\ b_3b_2b_1b_0 \\ \hline c_4\ s_3s_2s_1s_0 \end{array} \tag{12.3}$$

式中,$s = s_3\,s_2\,s_1\,s_0$ 是 4 位的结果,而 c_4 是进位输出位。为了设计一个二进制字的加法器,把这个问题以列为基础分解成一位的加法器。在标准的进位计算中,i 列($i = 0,1,2,3$)中的每一列按照**全加器**方程操作:

$$\begin{array}{r} c_i \\ a_i \\ +\ \ b_i \\ \hline c_{i+1}\ \ s_i \end{array} \tag{12.4}$$

式中,c_i 是来自($i-1$)列的进位输出位,而 c_{i+1} 是该列的进位输出位。这一操作是由图 12.4 的全加器表格及简单的符号图来描述的。这个电路的最通常的表达式为



a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

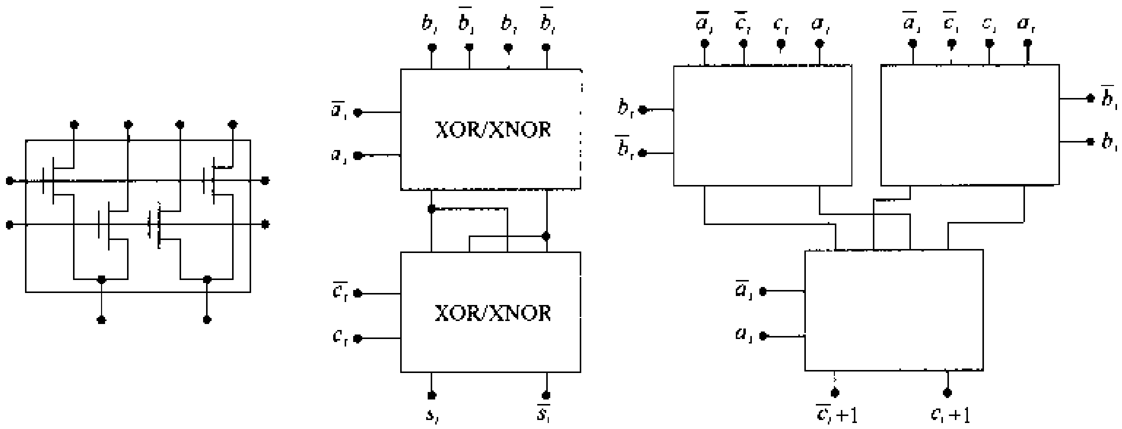
图 12.4 全加器符号及功能表

$$s_i = a_i \oplus b_i \oplus c_i$$
$$c_{i+1} = a_i \cdot b_i + c_i \cdot (a_i \oplus b_i) \qquad (12.5)$$

上式可从功能表的 SOP(积项之和)分析中直接推导出来。进位输出位也可以写成另外一种形式:

$$c_{i+1} = a_i \cdot b_i + c_i \cdot (a_i + b_i) \qquad (12.6)$$

一个特别简洁的电路实现是运用双轨互补传输管逻辑(CPL)。基本的功能块是 CPL 两输入阵列,它的一般形式显示在图 12.5(a)中。求和电路显示在图 12.5(b)中;第一个 XOR/XNOR 门的输出为以下一对信号:



(a) 2 输入阵列　(b) 求和电路　(c) 进位电路

图 12.5　CPL 全加器设计

$$a_i \oplus b_i \quad 及 \quad (\overline{a_i \ominus b_i}) \qquad (12.7)$$

第二个门产生全加器的和:

$$s_n = (\overline{a_i \oplus b_i}) \cdot c_i + (a_i \oplus b_i) \cdot \bar{c}_i \qquad (12.8)$$

图 12.5(c)中的进位电路采用两输入的阵列作为一个组合逻辑的基本单元。例如左上的阵列产生以下的输出:

$$\bar{a}_i \cdot b_i + \bar{b}_i \cdot \bar{c}_i$$
$$\bar{b}_i \cdot c_i + a_i \cdot b_i \qquad (12.9)$$

而右上的电路产生:

$$\bar{a}_i \cdot \bar{b}_i + b_i \cdot \bar{c}_i$$
$$b_i \cdot c_i + a_i \cdot \bar{b}_i \qquad (12.10)$$

最后一个门运用这些结果产生 ci_{+1}和 $\bar{c}_{i+1}$。尽管这是一个看来简单的设计,但必须记住 CPL 是一个双轨电路,它在每级都要求互补的一对变量如(a_i,$\bar{a}_i$)。同时当逻辑 1 的电压通过一个 nFET 时,阈值电压损失会使它的值降低,所以输出端需要电平恢复缓冲器或锁存电路。因此 CPL 多半是实现 CMOS 全加器的一种特定设计。CPL 半加器是很容易构成的,因为它只需要 XOR/XNOR 和 AND/NAND 功能。

全加器的行为描述可以通过简单修改半加器的模型而得到:

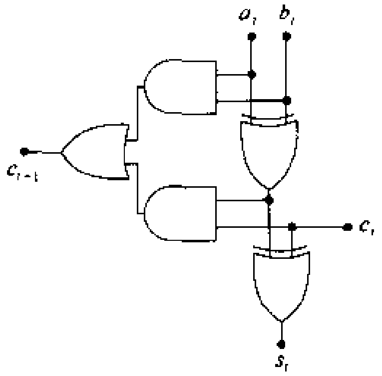
```
module full_adder(sum,c_out,a,b,c_in);
```

```
input a,b,c_in;
output sum,c_out;
assign {c_out,sum} = a+b+c_in;
endmodule
```

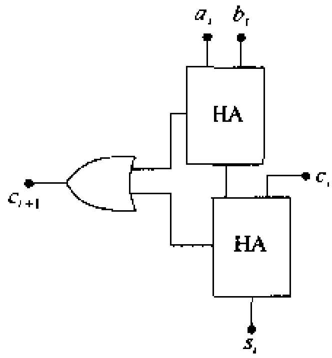
所有变量都是标量(单个二进制位),而链接算子则产生两个输出。结构模型可以用图 12.6(a)所示的门级电路为基础,这是方程组一对一的直接转换。在这一层次上,模型的形式如下:

```
module full_adder_gate(sum,c_out,a,b,c_in);
input a,b,c_in;
output sum,c_out;
wire w1,w2,w3;
xor(w1,a,b),
   (sum,w1,c_in);
and(w2,a,b),
   (w3,w1,c_in);
or(c_out,w2,w3);
endmodule
```

这里采用了稍有不同的变量标识符以使代码的可读性较好。一个全加器也可以用两个半加器(HA)模块来构成,如图 12.6(b)所示。用下面定义的模块进行例举。



(a) 门级逻辑



(b) 基于半加器的设计

图 12.6 全加器逻辑电路

```
module half_adder_gate(sum,c_out,x,y);
...
```

可以得到以下的描述:

```
module full_adder_HA(sum,c_out,a,b,c_in);
input a,b,c_in;
output sum,c_out;
wire wa,wb,wc;
half_adder_gate(wa,wb,a,b);
half_adder_gate(sum,wc,wa,c_in);
or(c_out,wb,wc);
endmodule
```

由于全加器的重要性,多年来开发了好几种。静态 CMOS 逻辑电路的 AOI 实现可以通过方程(12.6)写出进位输出而得到。这能写成:

$$\overline{s_i} = (a_i + b_i + c_i) \cdot \bar{c}_{i+1} + (a_i \cdot b_i \cdot c_i) \qquad (12.11)$$

所以 c_{i+1}和 $\bar{s}_i$ 均为 SOP(积项之和)形式。而且 $\bar{s}_i$ 利用了 $\bar{c}_{i+1}$，可以设计一个 AOI 门来得到 $\bar{c}_{i+1}$，并把这个输出送入另一个 AOI 门以求得 $\bar{s}_i$。图 12.7 为所构成的两个 OAOI 电路。上面的电路产生 $\bar{c}_{i+1}$而下面的电路经反向后得到 s_i。用串－并联 CMOS 门设计这两个 OAOI 电路是很容易的。然而注意方程(12.11)包含 4 个 OR 操作，这表明下面的 AOI 门含有 4 个串联的 pFET。这也许会在一个字加法器中引起无法接受的长延时。

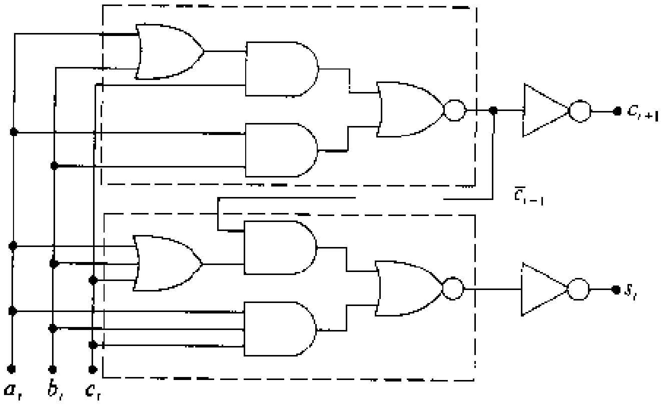
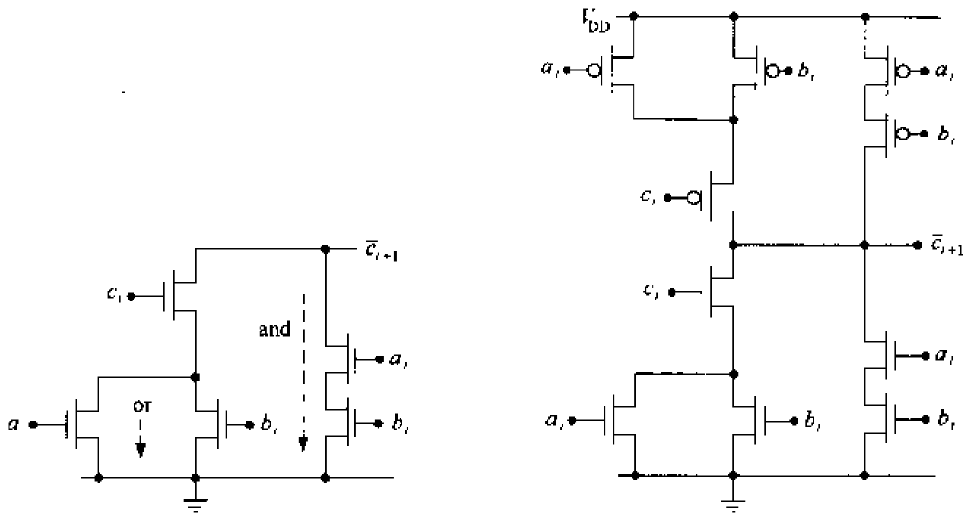


图 12.7　AOI 实现的全加器逻辑电路

为设计更有效的电路，考虑由 AOI 逻辑图表示的进位输出电路的 nFET 阵列实现。采用标准结构得到图 12.8(a)的 nFET 电路，可以看到相应于以下两项有两个主要的下拉路径：

$$a_i \cdot b_i$$
$$c_i \cdot (a_i + b_i) \tag{12.12}$$

如果上面两项中任何一项为 1，那么输出就被下拉至 0(接地)。当 a_i 和 b_i 同时为 1 时，AND 项起作用；当 $c_i = 1$ 时，无论 $a_i = 1$ 还是 $b_i = 1$，OR 项都形成下拉路径。这使我们可以构成一个 pFET 的镜像电路，得到如图 12.8(b)所示的整个门。如 $a_i = b_i = 0$，串联的 pFET 则上拉至 1(V_{DD})，这恰好与 nFET 下拉的条件相反。如果只有一个为 0，那么输出就由 c_i 的值来决定。



(a) 标准 nFET 逻辑　　(b) 镜像电路

图 12.8　进位输出电路的改进

为了完成镜像 CMOS 全加器的设计，让我们把全加器的和 $\bar{s}_i$ 写成简单的 OR 门如下：

$$\bar{s}_i = A + B \tag{12.13}$$

式中

$$\begin{aligned} A &= (a_i + b_i + c_i) \cdot \bar{c}_{i+1} \\ B &= (a_i \cdot b_i \cdot c_i) \end{aligned} \tag{12.14}$$

这与进位输出电路具有相同的特性，但可构成如图 12.9 所示的完整的全加器。这个电路比串并联的实现要快，且由于是镜像的 FET 阵列，所以易于画成版图。

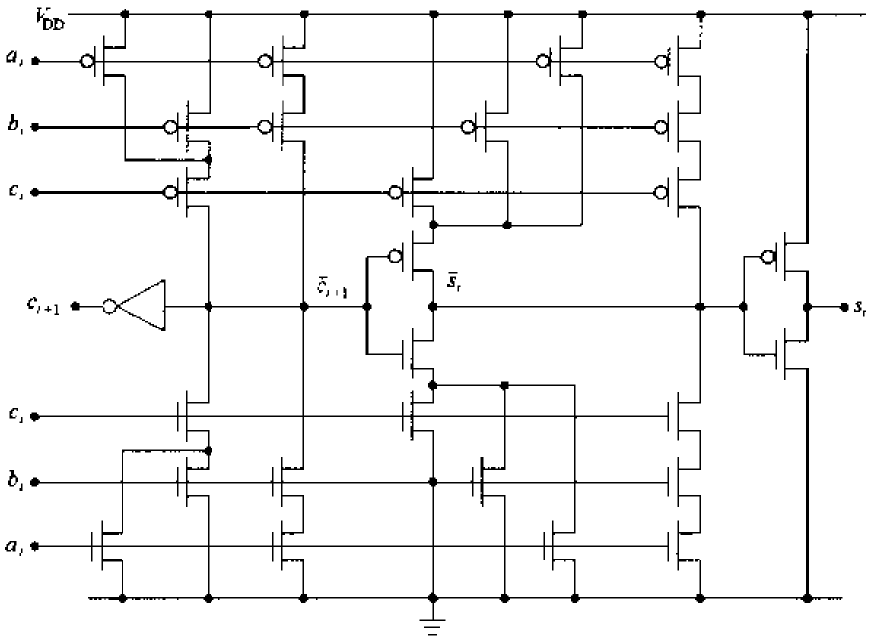


图 12.9 镜像 AOI 实现的 CMOS 全加器

图 12.10 为基于传输门(TG)的全加器。输入电路提供 XOR 和 XNOR 操作，它们的结果由 TG 的输出阵列用于产生“和”与“进位”。这个电路的特点是 s_i 和 c_{i+1} 的延时几乎相同，这通过跟踪经上面和下面部分的逻辑信号流路径可以很容易地看出。如果输入的各位同时加入，那么和位和进位输出将在几乎相同的时间同时产生。这与 AOI 实现的电路有明显的不同，后者首先产生进位输出，然后利用它再计算“和”。

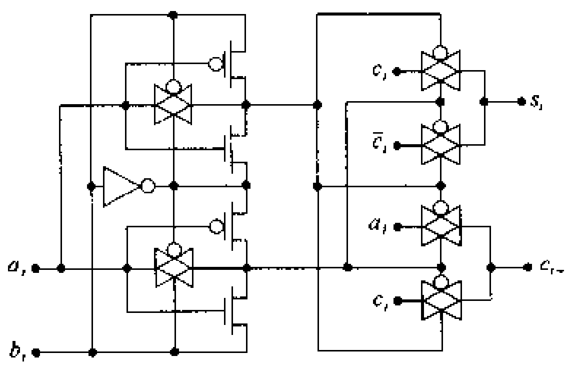


图 12.10 传输门实现的全加器电路

12.2 串行进位加法器

既然有了加单个二进制位的基础，就让我们把这个问题扩展到加两个二进制字。一般地，将两个 n 位的字相加产生一个 n 位的和及一个进位输出位 c_n，后者可以作为另一个更高位加法器的进位输入，或者作为一个溢出标志。图

12.11 为它的一般符号。在最初的讨论中将运用 $n=4$。

串行进位加法器是基于加法方程

$$\begin{array}{r} c_3\ c_2\ c_1\ c_0 \\ +\ a_3a_2a_1a_0 \\ +\ b_3b_2b_1b_0 \\ \hline c_4\ s_3s_2s_1s_0 \end{array} \tag{12.15}$$

式中，c_i 代表从前一列来的进位输入位。不失一般性，将保留第 0 位的进位输入位为 c_0。

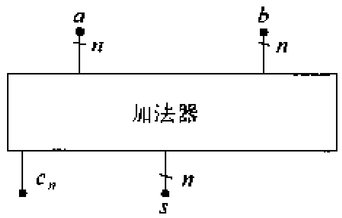


图 12.11 一个 n 位加法器

注意若包括进位输入字，这实际上是相加三个二进制字。一个 n 位的串行进位加法器要求 n 个全加器，其中进位输出位 c_{i+1} 用来作为下一列的进位输入。图 12.12 为一个 4 位字的串行进位加法器的情形。

高层次的模型可以用 Verilog 向量来建立，如以下的代码所示：

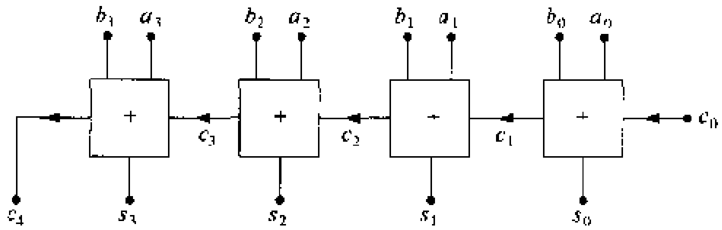


图 12.12 4 位串行进位加法器

```
module four_bit_adder(sum,c_4,a,b,c_0);
input[3:0]a,b;
input c_0;
outut c_4;
assign {c_4,sum} = a + b + c_0;
endmodule
```

上面的代码用链接算子来产生一个 5 位的结果，它包含“和”以及进位输出位 c_4。另一种模拟它的方法是采用 4 个全加器模块，按所画的那样把它们用导线连接在一起：

```
module FA_module(sum,c_4,a,b,c_0);
input[3:0]a,b;
input c_0;
output[3:0]sum;
output c_4;
wire c_1,c_2,c_3;
/* 以下例举的一位 FA 模块的句法为
        full_adder(sum,c_out,a,b,c_in) */
full_adder fa0(sum[0],c_1,a[0],b[0],c_0);
full_adder fa1(sum[1],c_2,a[1],b[1],c_1);
full_adder fa2(sum[2],c_3,a[2],b[2],c_2);
full_adder fa3(sum[3],c_4,a[3],b[3],c_3);
endmodule
```

这个例子采用以下表示方法来定义一个向量的第 i 位：

```
sum[i],a[i],和 b[i]
```

这是很容易理解的。如果定义一个量：

```
input[7:0]Q;
```

若 Q=10001110,那么 Q[0]=0,Q[1]=1,Q[2]=1,以此类推。而且我们总是假设,如

```
full_adder(sum,c_out,a,b,c_in);
```

这样被例举的模块已定义在代码的其他地方。它们可以在任何层次上,写出从行为描述向下到门级的结构描述。

串行进位加法器的结构易于连接相邻的电路。正是由于这一特点,使这个设计的速度较慢。因为任何一个全加器只有当它的输入进位位有效时其输出才有效,最左边的电路是最后响应的。只有在此时,全加器输出字的结果才有效。

整个延时取决于全加器电路的特性。不同的 CMOS 实现将产生不同的最坏情形的延时路径。为了说明,假设在 4 位加法器电路中运用图 12.9 的 AOI 镜像 CMOS 全加器。由于需要用进位输出来计算"和",因此使从 c_i 到 c_{i+1} 的进位延时减到最小。图 12.13 显示了这个加法器最长的延时路径,即进位传送通过每一级;假设所有的输入同时有效,相加各个延时以得到如下的总延时 t_{4b}:

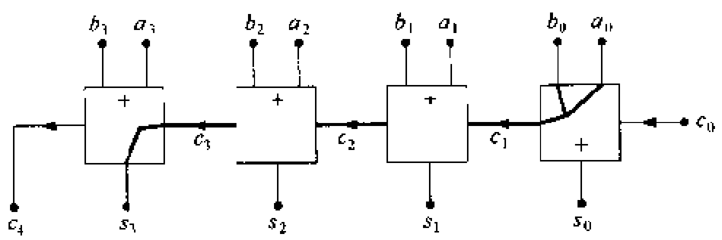


图 12.13 通过 4 位串行加法器最坏情形下的延迟

$$t_{4b} = t_{d3} + t_{d2} + t_{d1} + t_{d0} \tag{12.16}$$

式中,t_{di} 是通过第 i 级的最坏情形下的延时。可以求出每级的延时。对于第 0 位,$t_{d0}=t_d(a_0, b_0 \to c_1)$,这是由输入产生进位输出位的时间。通过第 1 和第 2 个全加器的延时相同,即从进位输入到进位输出的时间:$t_{d1}=t_{d2}=t_d(c_{in} \to c_{out})$。在本设计中最后第 3 级的延时是产生输出"和"位 s_3 的时间,可以写为 $t_{d3}=t_d(c_{in} \to s_3)$。于是总的延时为

$$t_{4b} = t_d(c_{in} \to s_3) + 2t_d(c_{in} \to c_{out}) + t_d(a_0, b_0 \to c_1) \tag{12.17}$$

如果将此扩展到一个 n 位的串行进位加法器,那么最坏情形下的延时为

$$t_{n-bit} = t_d(c_{in} \to s_{n-1}) + (n-2)t_d(c_{in} \to c_{out}) + t_d(a_0, b_0 \to c_1) \tag{12.18}$$

这表明延时与 n 成正比。可用符号表示为

$$\text{delay} \sim O(n) \tag{12.19}$$

因此串行结构对于较大的字长不是一个很好的选择。

在进一步介绍较为先进的加法器设计之前,让我们回想一下一个补码减法器可以通过增加 XOR 门以及一个 add_sub 的控制位来实现,如图 12.14 所示。当 add_sub=0 时,XOR 门使 b_i 位通过,输出为"和"$(a+b)$。当控制位 add_sub=1 时,XOR 变为反相器,于是反码值

b_i 进入全加器；add _ sub = 1 也起到进位输入 $c_0 = 1$ 的作用。这两个操作合在一起就得到“差”$(a-b)$的补码。这一技术也可有限度地运用到其他加法器电路中。

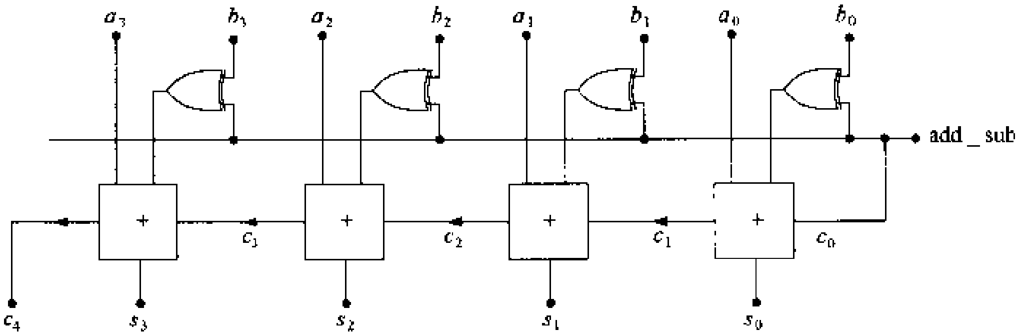


图 12.14　4 位加法器-减法器电路

12.3　超前进位加法器

超前进位(CLA)加法器设计用来克服由于串行进位影响引起的时间滞后(latency)。CLA 算法来源于如下产生进位输出位的方程：

$$c_{i+1} = a_i \cdot b_i + c_i \cdot (a_i \oplus b_i) \qquad (12.20)$$

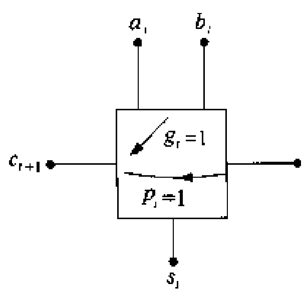
上式也就是使 $c_{i+1} = 1$ 的情形。因为式中每项都可以使这个输出为 1，我们分别来处理这两项。首先如果 $a_i \cdot b_i = 1$，　那么 $c_{i+1} = 1$。称

$$g_i = a_i \cdot b_i \qquad (12.21)$$

为进位产生，因为这两个输入可以看成“**产生**”了进位输出位。注意如果 $g_i = 1$，那么必有 $a_i = b_i = 1$。第二项代表输入进位 $c_i = 1$ 可以通过全加器“传播”的情形。这发生在当“**传播**”项

$$p_i = a_i \oplus b_i \qquad (12.22)$$

等于 1 时；如果 $p_i = 1$，那么 $g_i = 0$，因为 XOR 操作只有在两个输入不相等的时候才产生 1。图 12.15 显示了进位产生和进位传播项的工作情况。根据这两项的定义，进位输出位的方程为



	g_i	p_i
	$a_i \cdot b_i$	$a_i \oplus b_i$
$a_i = b_i = 0$	0	0
$a_i = b_i = 1$	1	0
$a_i \neq b_i$	0	1

$c_{i+1} = a_i \cdot b_i + c_i \cdot (a_i \oplus b_i)$

图 12.15　超前进位算法的基础

$$c_{i+1} = g_i + p_i \cdot c_i \qquad (12.23)$$

CLA 的主要概念是先对每位计算 p_i 和 g_i 的值，然后运用它们来计算进位 c_{i+1}。一旦这些算出，那么每个 i 的“和”可由下式给出

$$s_i = p_i \oplus c_i \qquad (12.24)$$

这避免了进位串行地传播通过进位链。

让我们来分析 4 位的 CLA 方程。假设 c_0 已知，有

$$c_1 = g_0 + p_0 \cdot c_0 \tag{12.25}$$

c_2，c_3 和 c_4 的表达式具有相同的形式，即

$$\begin{aligned} c_2 &= g_1 + p_1 \cdot c_1 \\ c_3 &= g_2 + p_2 \cdot c_2 \\ c_4 &= g_3 + p_3 \cdot c_3 \end{aligned} \tag{12.26}$$

这些可以用基本的进位产生和进位传播项来表示，因为注意到 c_i 可以连续地替换 c_{i+1}。第一个简化是把 c_1 代入 c_2 的方程得到：

$$c_2 = g_1 + p_1 \cdot (g_0 + p_0 \cdot c_0) \tag{12.27}$$

将它展开得到：

$$c_2 = g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot c_0 \tag{12.28}$$

类似地，将 c_2 代入 c_3 得到：

$$\begin{aligned} c_3 &= g_2 + p_2 \cdot (g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot c_0) \\ &= g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot c_0 \end{aligned} \tag{12.29}$$

最后，进位输出位为：

$$\begin{aligned} c_4 &= g_3 + p_3 \cdot (g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot c_0) \\ &= g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 + p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0 \end{aligned} \tag{12.30}$$

这些方程表明，每个进位位可从进位产生和进位传播项求出。而且这一算法得到了嵌套的 SOP(积项和)的表达式。运用展开的表达式，4 位电路的逻辑图显示在图 12.16 中。注意这一逻辑门电路的结构特点。一旦算出进位输出位，那么运用方程(12.24)中简单的 XOR 门就可以算出“和”。完整的加法器电路显示在图 12.17 中。这里“CLA 进位链”方块代表图 12.16 中的进位逻辑。这表明它与串行进位设计有显著的不同。

4 位加法器的 Verilog 高层次抽象描述可以描述任何加法器，包括基于 CLA 的设计。然而，可以重写行为代码以明显的方式较好地显示内部的算法。以下基于 assign 语句的 RTL 模型说明了这个概念。

```
module CLA_4b(sum,c_4,a,b,c_0);
input[3:0]a,b;
input c_0;
output[3:0]sum;
output c_4;
wire p0,p1,p2,p3,g0,g1,g2,g3;
wire c1,c2,c3,c4;
assign
      p0 = a[0]^b[0],
      p1 = a[1]^b[1],
      p2 = a[2]^b[2],
      p3 = a[3]^b[3],
      g0 = a[0]& b[0],
      g1 = a[1]& b[1],
      g2 = a[2]& b[2],
      g3 = a[3]& b[3];
```

```
assign
        c1 = g0 | (p0 & c_0),
        c2 = g1 | (p1 & g0) | (p1 & p0 & c_0),
        c3 = g2 | (p2 & g1) | (p2 & p1 & g0) | (p2 & p1 & p0 & c_0),
        c4 = g3 | (p3 & g2) | (p3 & p2 & g1) | (p3 & p2 & p1 & g0)
            | (p3 & p2 & p1 & p0 & c_0);
assign
        sum[0] = p0 ^ c_0,
        sum[1] = p1 ^ c1,
        sum[2] = p2 ^ c2,
        sum[3] = p3 ^ c3,
          c_4 = c4;
endmodule
```

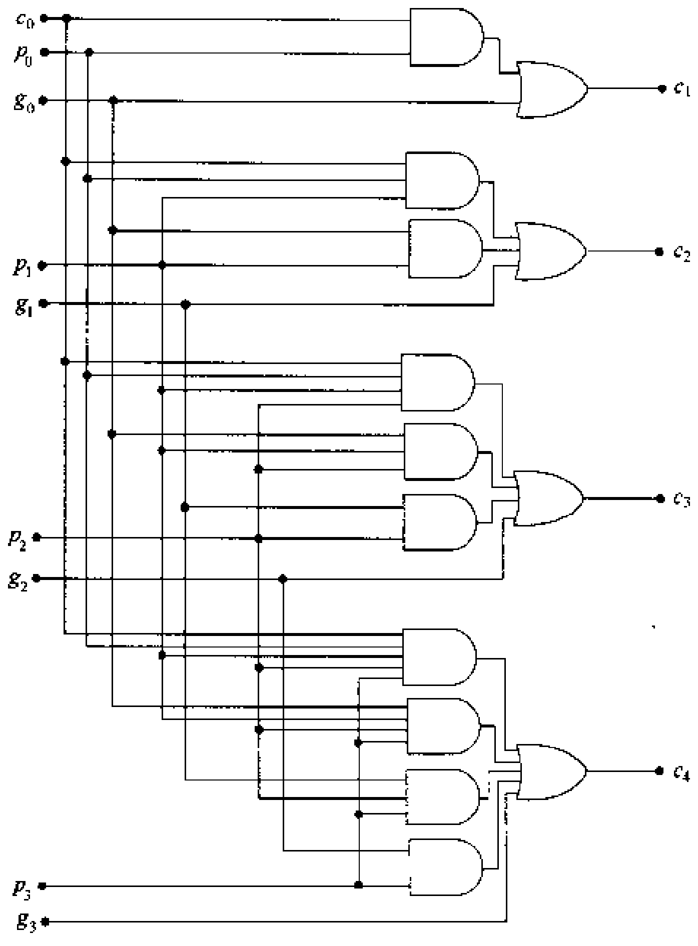


图 12.16　4 位 CLA 进位的逻辑电路

把延时加到每个语句中则得到模拟所需要的最终信息。CLA 方程的这个重复特点通过运用 Verilog"for"过程可以用一种更为有效的编码方式来实现。

为了把 CLA 算法转换成电路，采用第 2 章中介绍的逻辑构成技术来产生如图 12.18 中所示的 nFET 阵列。注意，每个进位输出电路 $\bar{c}_i$ 形成下一个较高位进位 $\bar{c}_{i+1}$ 的基础。这是由于该算法的嵌套特点所致。

一旦设计了 nFET 逻辑，它就可以用在各种各样的电路中。图 12.19 为三种可能的电路。图 12.19(a)的结构是一个标准的互补结构，这里我们采用小圆圈推移方法构成 pFET 阵列得到了串并联的 pFET 阵列。也可以采用图 12.19(b)的静态准 nMOS 方法，但必须考虑器件尺寸比以保证输出低电压 V_{OL} 足够小而又不必采用尺寸过分大的 nFET。为了避免这一点，可以采用动态逻辑系列，如图 12.19(c)所示。然而这会引起时序问题，并且其输出只在一段很短的时间内有效。显然选择什么样的电路系列涉及到许多考虑因素。

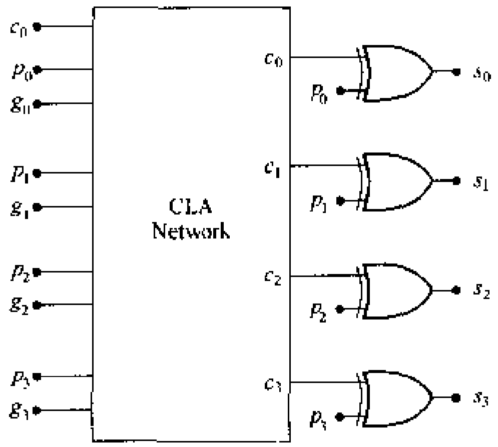
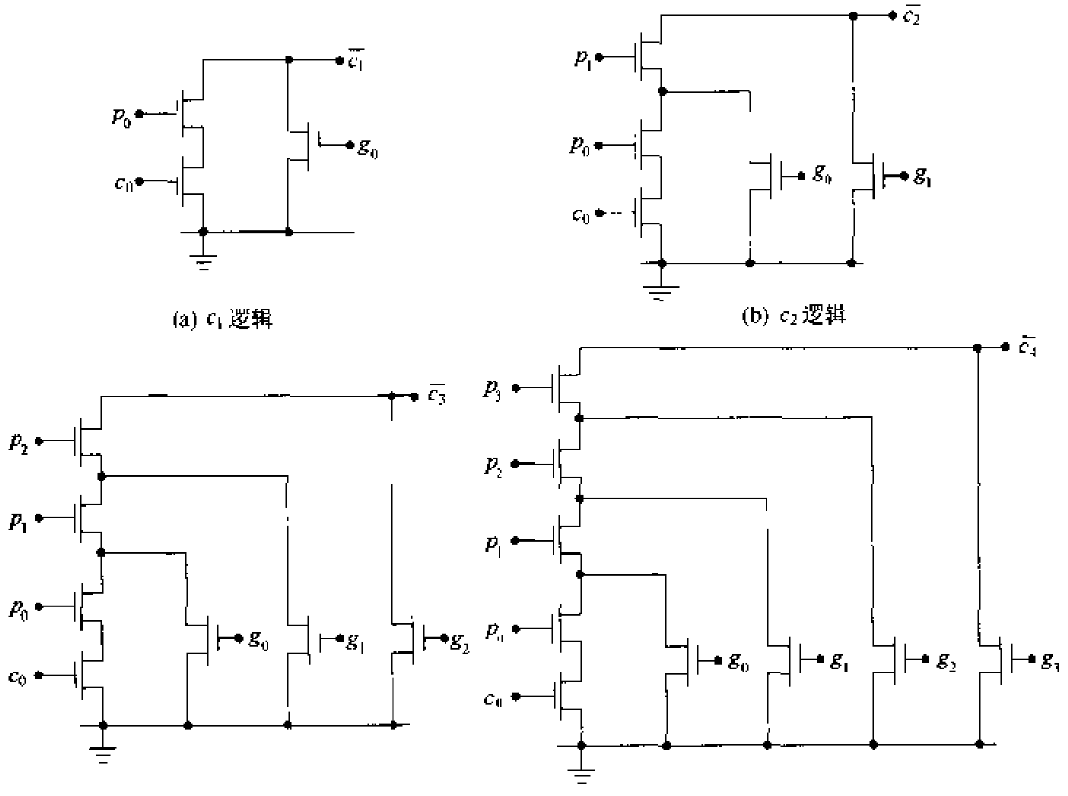


图 12.17　用 CLA 进位电路计算和



(a) c_1 逻辑　(b) c_2 逻辑

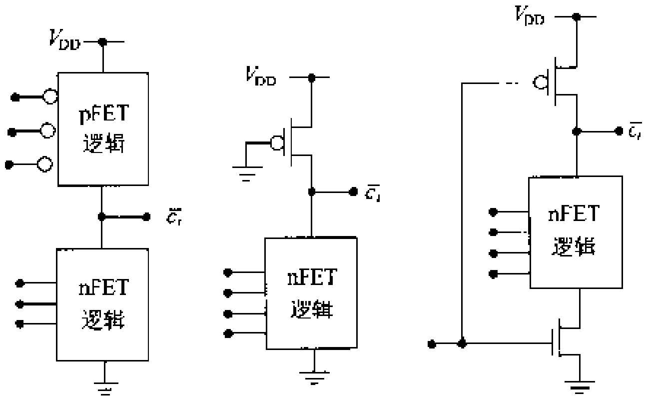
(c) c_3 逻辑　(d) c_4 逻辑

图 12.18　实现 CLA 各项进位的 nFET 逻辑阵列

让我们考察一下采用全互补静态电路的可能性。图 12.20(a)的 $\bar{c}_1$ 电路显示了具有串并联结构的完整的 nFET/pFET 阵列。然而由于它的形式类似于前面在图 12.8 中分析的进位输出电路，我们可以构成如图 12.20(b)所示的镜像等效逻辑门。这一过程可以继续应用到其余各位。例如图 12.21 为 $\bar{c}_2$ 的镜像电路。注意在阵列中的对称性。这一特点可以在物理设计层次上实现比较结构化的版图。

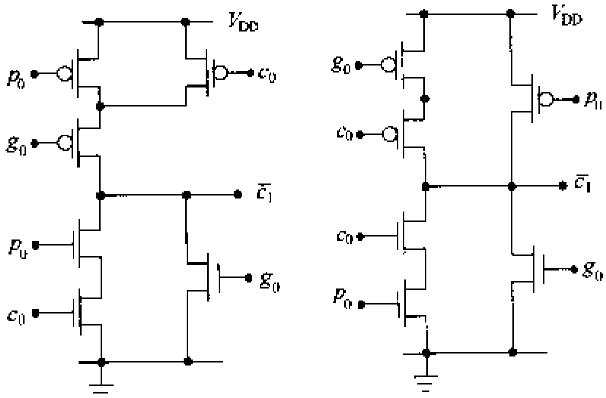
另一种方法是基于运用多个输出的多米诺逻辑(MODL)。这之所以可能是因为进位从一位到下一位的嵌套特点提供了实现 MODL 所需要的"与"的关系。为了分析这一点，回想一下

我们曾得到过



(a) 互补 FET (b) 准 nMOS (c) 动态

图 12.19 图 12.18 中 nFET 逻辑阵列可能采用的三种电路



(a) 串并联电路 (b) 镜像等效电路

图 12.20 静态 CLA 镜像电路

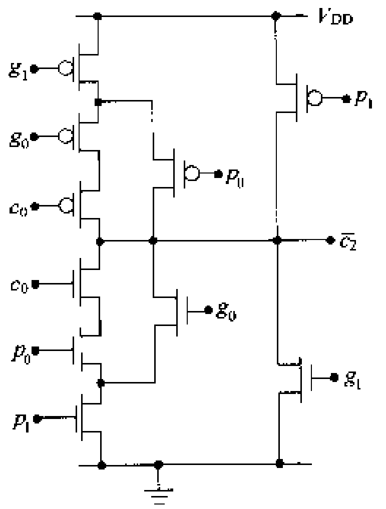


图 12.21 产生 $\bar{c}_2$ 的静态镜像电路

$$
\begin{aligned}
c_1 &= g_0 + p_0 \cdot c_0 \\
c_2 &= g_1 + p_1 \cdot c_1 \\
&= g_1 + p_1 \cdot (g_0 + p_0 \cdot c_0)
\end{aligned}
\tag{12.31}
$$

可用 c_1 作为一个输出，而 c_2 为另一个输出，这两者是由"与"操作联系在一起的。因为这些关系对 c_3 和 c_4 同样成立，所以只需要单个 MODL 门就可以产生所有四个进位位。而且 MODL 是非反相位的逻辑系列，反相器已属于它结构本身的一部分。图 12.22 为一个 4 位的 MODL 进位电路，其中逻辑阵列为每个进位位提供了各自的输出。每个内部节点附有一个预充电 pFET。当时钟 $\phi=0$ 电路处于预充电时，所有的输出都被驱动至 0。该逻辑电路在求值阶段 ($\phi=0$)时接受输入。如果 c_1 电路允许内部节点放电而产生 $c_1=1$，那么必有以下两个条件之一成立：

$$g_0 = 1$$

或 (12.32)

$$p_0 \cdot c_0 = 1$$

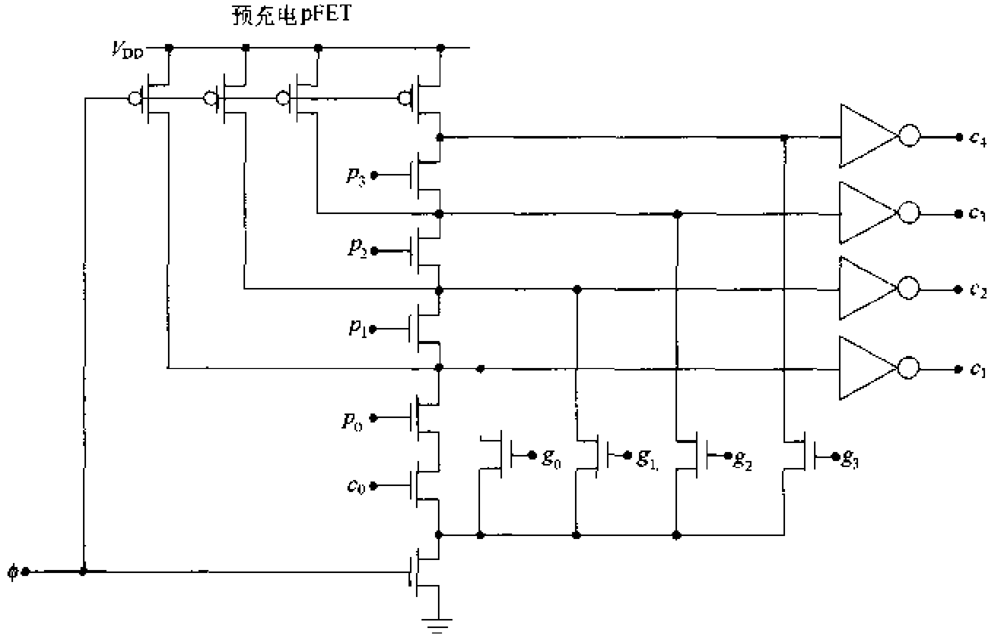


图 12.22 MODL 进位电路

如果 $p_0 \cdot c_0 = 1$，那么 $p_1 = 1$ 将驱动 c_2 至 1。另一种可能是进位输出 c_2 由 $g_1 = 1$ 产生。这类相互依赖关系在逻辑电路中继续向高位进行以产生 c_3 和 c_4。用一个逻辑门产生 4 个进位输出位的功能是非常吸引人的。版图的复杂程度相当于单个的 c_4 镜像门再加上几个管子。然而应当记住，MODL 是一种动态电路技术，因此它具有通常的局限：必须有时钟，输出会有电荷泄漏以及电荷分享，而且串联的 nFET 链会产生很长的放电时间，除非采用尺寸很大的 FET。

12.3.1 曼彻斯特进位链

曼彻斯特进位技术对产生 CLA 进位是一种特别出色的方法。它是以建立实现如下基本方程的开关逻辑电路为基础的：

$$c_{i+1} = g_i + p_i \cdot c_i \tag{12.33}$$

这一电路可以串接起来送入后续的各级中。

考虑一个全加器，输入为 a_i，b_i 和 c_i。我们将运用“产生”和“传播”表达式

$$\begin{aligned} g_i &= a_i \cdot b_i \\ p_i &= a_i \oplus b_i \end{aligned} \tag{12.34}$$

并引入**进位清除**位 k_i：

$$\begin{aligned} k_i &= \overline{a_i + b_i} \\ &= \bar{a}_i \cdot \bar{b}_i \end{aligned} \tag{12.35}$$

这一项之所以称为“进位清除”是基于这样的事实，即如 $k_i = 1$，则 $p_i = 0$ 和 $g_i = 0$，所以 $c_{i+1} = 0$；因此 $k_i = 1$ 则“清除”了进位输出位。这可以从图 12.23 的表格中得到验证，表格对于所有可能的输入列出了 p_i，g_i 和

a_i	b_i	p_i	g_i	k_i
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	0	1	0

图 12.23 进位传播、进位产生和进位清除

k_i 的值。注意对于给定的一组输入(a_i,b_i),这三个量中只有一个是逻辑 1。

曼彻斯特进位技术就是基于这个特点。因为三个量 p_i,g_i 和 k_i 中只有一个量可以为 1,可用图 12.24 所示的 FET 来构成开关电路。所选择的拓扑连接使每一时刻只有一个 FET 是闭合的。这个操作可以通过考察每种可能性来理解。首先若有$(a_i,b_i)=(0,0)$,那么 $k_i=1$ 及 $c_{i+1}=0$。如果 $a_i \neq b_i$,则 $p_i=1$,输入位 c_i 传播通过该电路使 $c_{i+1}=c_i$。最后,输入$(a_i,b_i)=(1,1)$意味着进位输出已由 $g_i=1$ 产生,所以 $c_{i+1}=1$。在电路级,应当注意仅用 nFET 会使当逻辑 1 传送通过管子时引起阈值电压降。

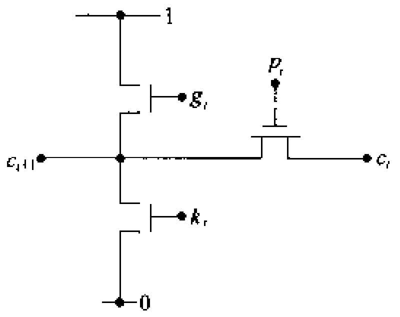


图 12.24 进位输出方程的开关电路

可以构成几个不同的曼彻斯特进位链。图 12.25 是其中的两个。图 12.25(a)中的静态逻辑门利用 $\bar{c}_i$ 作为输入。首先假设 $p_i=0$。这会断开 M1 并阻止输入 $\bar{c}_i$ 传播通过,但同时也使 nFET M3 导通。如果 $g_i=0$,pFET M4 导通使输出上拉到 $\bar{c}_{i+1}=1$。如果 $g_i=1$,那么 nFET M2 和 M3 同时导通而 M4 截止,使输出 $\bar{c}_{i+1}=0$。$p_i=1$ 的情形比较复杂。进位产生 g_i 必定为 0,所以 pFET M4 导通而 nFET 链的作用如同开路因为 M3 截止。输出于是由 $\bar{c}_i$ 控制。如果 $\bar{c}_i=1$,那么它就会传送到输出端且被保持,因为此时 pFET 连接到电源,所以 $\bar{c}_{i+1}=1$。如果 $\bar{c}_i=0$,则电路简化成由 M4 和 M1 构成的准 nMOS 反相器,且在输出处 $p_i=1$。[①] 为使低电平输出 $\bar{c}_{i+1}=0$ 的电压值较小,必须选择 nFET/pFET 的尺寸比足够大,以得到比较低的输出电压。

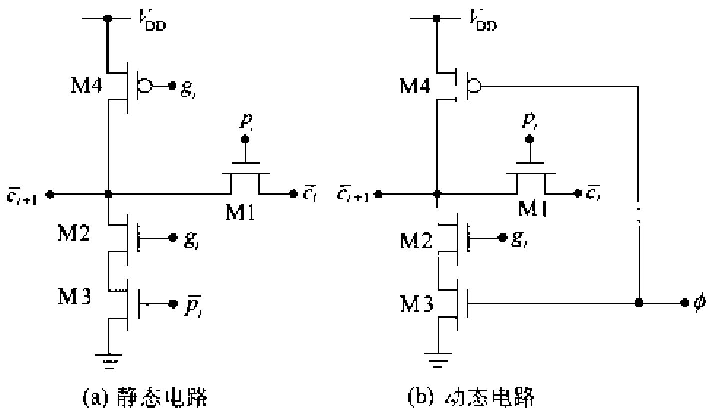


图 12.25 曼彻斯特电路

图 12.25(b)为一个动态电路。这个逻辑类似于静态设计,只是求值管 nFET M3 替代了原来的逻辑管。在预充电期间($\phi=0$),输出节点被拉至逻辑 1 电压。当时钟转换成 $\phi=1$ 时则开始求值。如果 $p_i=1$ 则发生进位传播,如果 $g_i=1$ 则该节点放电。这一电路可以用来构成图 12.26 所示的曼彻斯特进位链。当 $\phi=0$ 时每级都预充电。在求值期间产生各级的进位,其中 c_4 具有最长的时延。

① 使输入端 $\bar{c}_i$ 接地并遮住 M2-M3 管就很容易看到这一点。

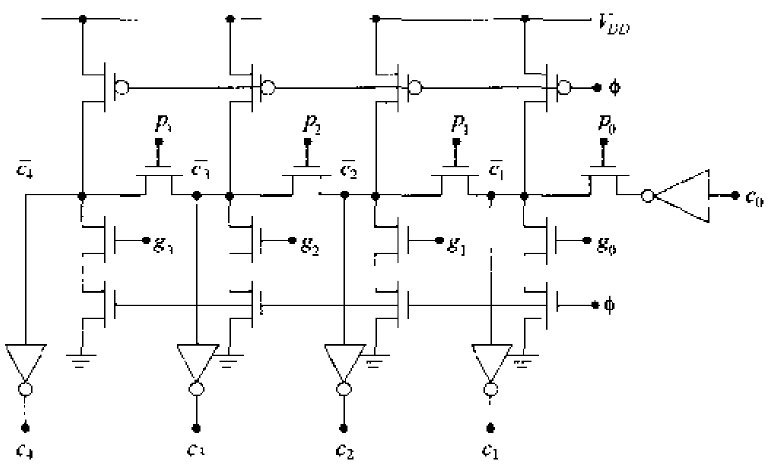


图 12.26　动态曼彻斯特进位链

12.3.2　扩展为宽位加法器

超前进位方程可以扩展到比 4 位更宽的加法器，必须注意由于通过最长时延路径的门数在增加而引起太长的硬件延时。如果完全不变地用这种方法设计一个 8 位的加法器，那么进位输出 c_8 就会具有如下形式的一项必须考虑：

$$p_7 \cdot p_6 \cdot p_5 \cdot p_4 \cdot p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0 \tag{12.36}$$

已经发表许多不同的技术来设计宽位加法器比较有效的 CLA 电路。考虑两个 n 位的字相加。Von Neumann 和其他人的研究表明最长进位链的平均长度为：②

$$\log_2(n) \tag{12.37}$$

例如一个 8 位加法器中的平均进位链长度为

$$\log_2(8) = \log_2(2^3) = 3 \tag{12.38}$$

而一个 32 位加法器的平均长度为

$$\log_2(32) = \log_2(2^5) = 5 \tag{12.39}$$

这意味着进位电路的长度没有必要扩展在整个字长上，可以被分隔成几个较小的部分。多层 CLA 电路就是基于这一原理的。

考虑图 12.27 的 n 位加法器的结构层：假设 $n = 2^k$，其中 k 为整数。选择一个位的位置 i，设它是 4 的倍数，对从 i 至 $i+3$ 位构成一个 4 位的**超前进位产生电路**。图 12.28 为这个电路的功能细节。它用“产生”和“传播”位来产生通常的进位输出位 c_{i+1}，c_{i+2}，和 c_{i+3}，但同时也计算**出块产生**信号 $g_{[i,i+3]}$和**块传播**信号 $p_{[i,i+3]}$，它们代表了 i 至 $i+3$ 这一组的整体特性并且能够送到加法器的更高位部分。图 12.29 的逻辑图提供了块产生和块传播信号的细节。注意它与图 12.16 中的 4 位 CLA 逻辑类似，差别在于块的输出电路，其连线不同。块产生信号可以用输入变量写成如下：

$$\begin{aligned} g_{[i,i+3]} = {} & g_{i+3} + p_{i+3} \cdot g_{i+2} + p_{i+3} \cdot p_{i+2} \cdot g_{i+1} \\ & + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot g_i \end{aligned} \tag{12.40}$$

② 见参考资料[4]。

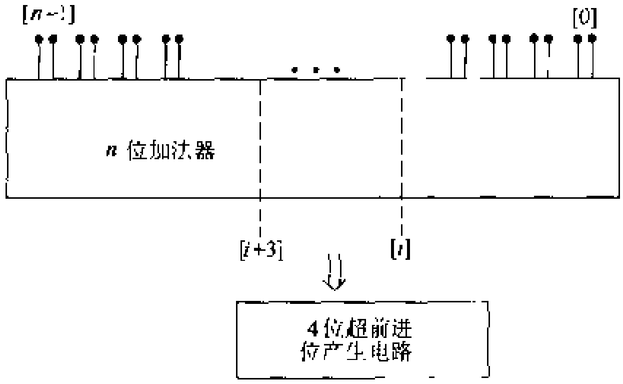


图 12.27 *n* 位加法器电路

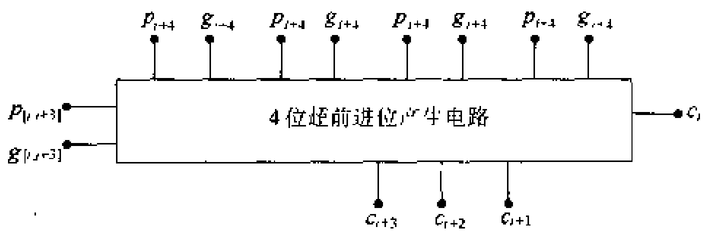


图 12.28 4 位超前进位产生电路的信号

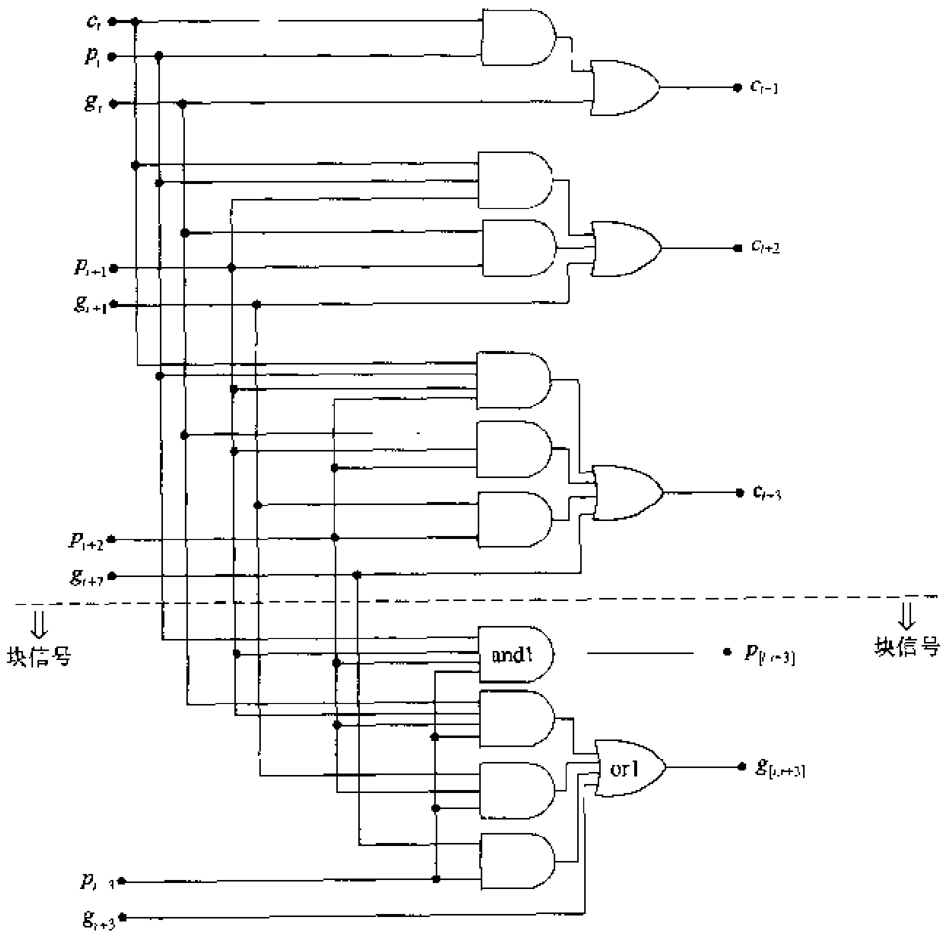


图 12.29 块超前进位产生逻辑

它取自图中标以 **or1** 的门。块传播为

$$p_{[i,i+3]} = p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i \qquad (12.41)$$

它是图中 **and1** 门的输出。块产生和块传播与一位的产生与传播是类似的，但它们提供了几位一组的整体特性。注意这个电路并没有计算最终的进位输出位 c_{i+4}。它也许需要也许不需要，这取决于加法器电路的整体结构。附加的逻辑可以按需要提供。

用多个超前进位产生块可以设计宽位加法器。一个例子是图 12.30 画出的 16 位的进位电路。输入 $a_{15}\ldots a_0$ 和 $b_{15}\ldots b_0$ 送入产生和传播电路，它们产生 CLA 块需要用的 $(p_{15}, g_{15}), \ldots (p_0, g_0)$ 值。CLA 子电路通常按层次进行描述。在层次 1，四个 4 位超前进位产生电路用来提供进位输出位 $c_{i+3}, c_{i+2}, c_{i+1}$ 以及块产生和块传播项 $g_{[i,i+3]}$ 和 $p_{[i,i+3]}$（$i = 0,4,8,12$）。块进位项然后送入层次 2 的一个 4 位超前进位电路。层次 2 的块产生进位输出位 c_4，c_8，c_{12} 以及字的产生和传播项 $g_{[0,15]}$ 和 $p_{[0,15]}$。此时所有的进位位除了 c_{15} 以外都已经计算出来，它们可以用于如下的求和方程中：

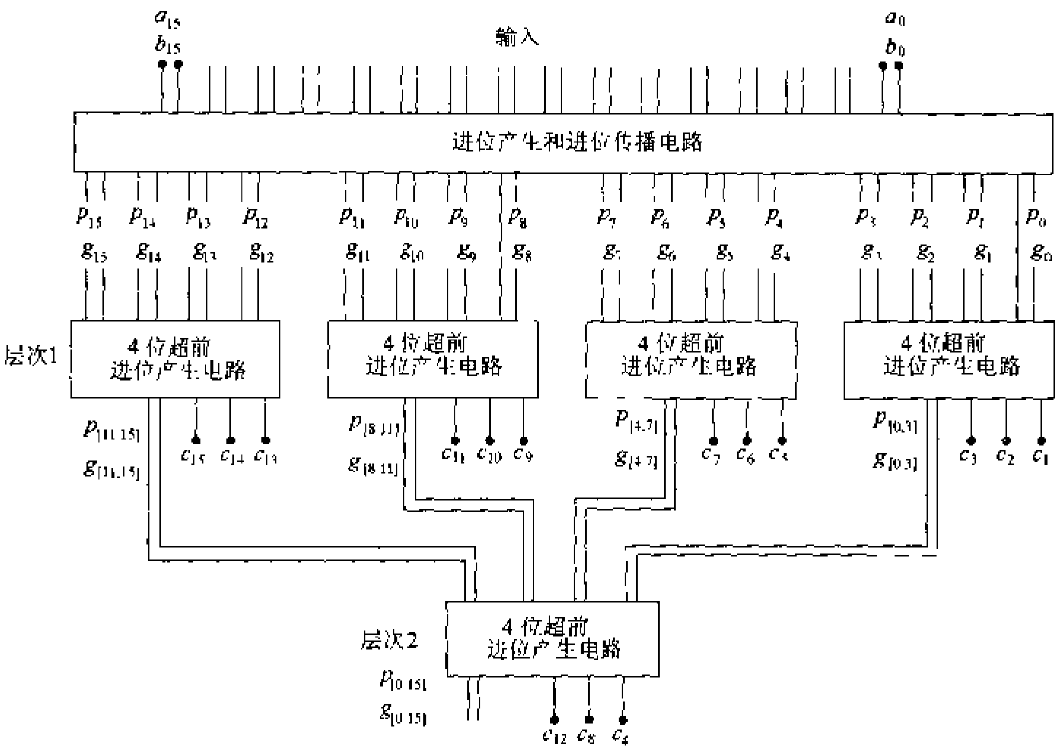


图 12.30 一个 16 位加法器的多层 CLA 块技术

$$s_i = p_i \oplus c_i \qquad (12.42)$$

对于一个 16 位加法器，最后的和位 s_{15} 和进位输出可以用字产生和字传播项计算出来。

64 位加法器可以通过对 16 位电路再增加另一层超前进位块来实现。这一技术显示在图 12.31 中。四个 16 位块产生四个 16 位一组的产生和传播项。这些项送入层次 3 的块，得到最终的进位输出位。应当注意每个块产生的进位输出位用于计算和。得到进位输出位的时间是随电路所在的层次而不同。层次 1 的位首先得到，层次 2 的位第二个得到，而层次 3 的位从电路中最终得到。并没有一个特别的理由一定要采用 4 位的超前进位产生电路，宽度较小或较大都是可以接受的。

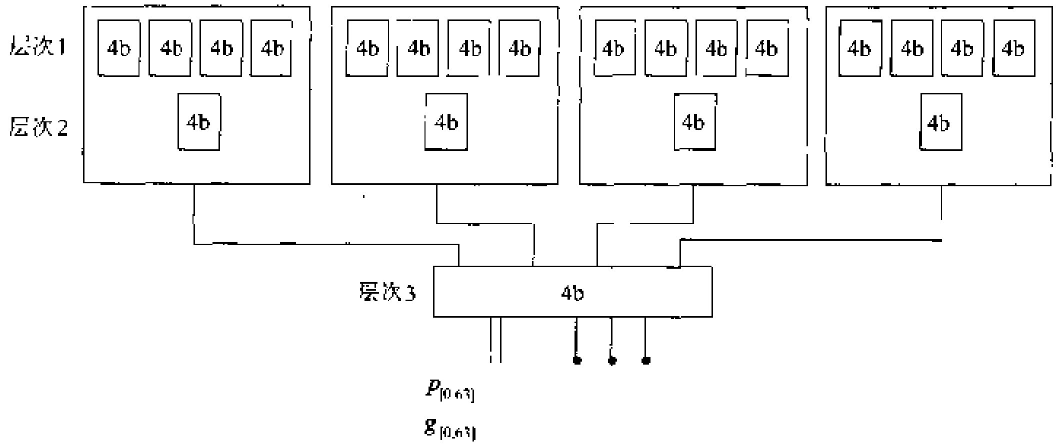


图 12.31 64 位 CLA 加法器结构

至此，已经考察了在 CLA 结构中涉及到的基本概念。感兴趣的读者可从参考资料[2]中看到更详细的讨论。

12.4 其他高速加法器

文献中发表了另外几种设计快速字加法器的方法。它们的目的都是为了减少计算时间，且每个都有不同的优缺点。本节考察几个这样的设计以说明在从高层次结构描述向下转变成电路时，可有许多不同的选择。

12.4.1 进位旁路电路(Carry-Skip Circuits)

进位旁路加法器设计加速一个宽位加法器，它使进位绕过整个加法器的一部分进行传播。对 4 位加法器的情形，这一概念显示在图 12.32(a)中。进位输入位表示成 c_i，而加法器本身产生一个进位输出位 c_{i+4}。进位旁路电路包括两个逻辑门。AND 门接受进位输入位并且将它与这一组的传播信号

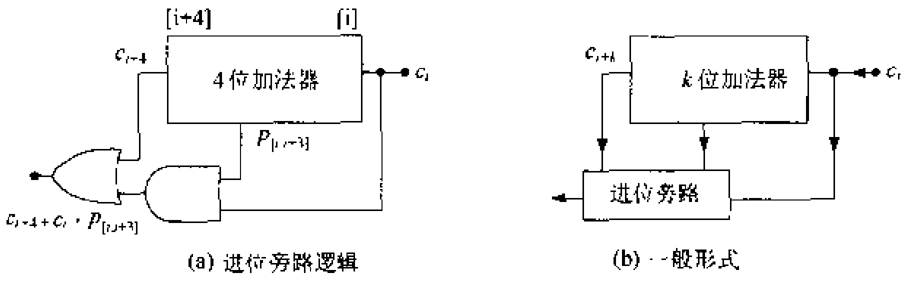


图 12.32 进位旁路电路

$$p_{[i,i+3]} = p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i \tag{12.43}$$

进行比较，上式中运用了各自的传播信号值。AND 门的输出与 c_{i+4}进行 OR 操作以产生这一级的输出

$$\text{carry} = c_{i+4} + p_{[i,i+3]} \cdot c_i \tag{12.44}$$

如图所示。如果 $p_{[i,i+3]}=0$,那么这一组的进位输出由 c_{i+4}的值来决定。如果当进位输入位 $c_i=1$ 时,$p_{[i,i+3]}=1$,那么这一组的进位输入就自动地送入加法器的下一组。"进位旁路"的名字来自如下的事实,即如果条件 $p_{[i,i+3]}\cdot c_i$ 为真,则进位输入位越过整个块。图 12.32(b)为 k 位一组时的一般情况。

图 12.33 是一个 16 位加法器进位旁路电路的例子。每个部分进位旁路组合的大小都选为 $k=4$。通过这一电路最坏情形下的延时是当 $c_0=0$ 并且第 0 位加法器产生一个进位输出位 $c_1=1$。如果采用的是串行进位加法器,那么最坏情况是这一位使 $c_4=1$,然后越过下两组[7,4]和[11,8]进入最后一个方块,最后串行通过该方块至输出端使 $c_{16}=1$。

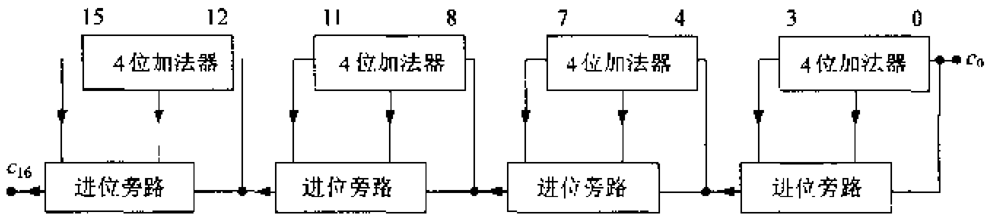


图 12.33 采用进位旁路电路的 16 位加法器

一个进位旁路块的尺寸 k 会影响这一设计的整体速度。已经证明,使一个 n 位加法器延时最小的最优旁路块的尺寸可以估计为

$$k = \sqrt{\frac{n}{2}} \tag{12.45}$$

对于 $n=16$,块的尺寸就是 $k=3$。另一种方法是采用可变的 k 值。进位旁路电路还可嵌套以构成多层次的电路。图 12.34 为一个两层进位旁路加法器的例子。

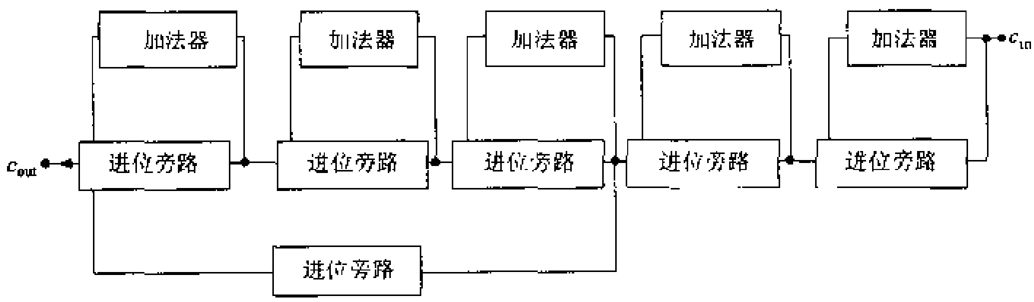


图 12.34 一个两层的进位旁路加法器

12.4.2 进位选择加法器(Carry-Select Adder)

进位选择加法器采用多个位数较少的加法器来构成快速的宽位加法器。考虑相加两个 n 位的数 $a=a_{n-1}\ldots a_0$ 和 $b=b_{n-1}\ldots b_0$。就每位而言,加法器的延时从最低有效位第 0 位起向上逐渐增加,第 $n-1$ 位要求最为复杂的逻辑。一个进位选择加法器把加法问题划分成较小的组。例如,把 n 位的问题分割成两组,每组($n/2$)位,然后特别注意高位的那一组,它相加 n 位字的$a_{n-1}\ldots a_{n/2}$部分和 $b_{n-1}\ldots b_{n/2}$部分。于是进位延时问题的中心就是由低位字部分 $a_{(n/2)-1}\ldots a_0$ 和 $b_{(n/2)-1}\ldots b_0$ 相加产生的进位输出位 $c_{n/2}$。对于这个进位位,只存在两种可能性:

$$c_{n/2}=0 \text{ 或 } c_{n/2}=1 \quad (12.46)$$

一个进位选择加法器对高位字部分提供了两个单独的加法器，每个加法器考虑一种可能性。然后用一个 MUX 选择正确的结果。

作为一个具体例子，考虑一个 8 位加法器分成两个 4 位一组。较低位 $a_3\ a_2\ a_1\ a_0$ 和 $b_3\ b_2\ b_1\ b_0$ 送入 4 位加法器 L 产生和位 $s_3\ s_2\ s_1\ s_0$ 以及一个进位输出位 c_4，如图 12.35 所示。较高位 $a_7\ a_6\ a_5\ a_4$ 和 $b_7\ b_6\ b_5\ b_4$ 作为这两个 4 位加法器的输入。加法器 U0 计算进位输入 $c=0$ 时的和，而 U1 完成同样的功能只是它的进位输入值 $c=1$。这两组结果作为一个 2:1 MUX 阵列的输入。从加法器 L 来的进位位 c_4 作为 MUX 的选择信号。如果 $c_4=0$，那么 U0 的结果送至输出端，如果 c_4 的值为 1 则选择 U1 的结果作为 $s_7\ s_6\ s_5\ s_4$。进位输出位 c_8 也是由 MUX 阵列选择的。

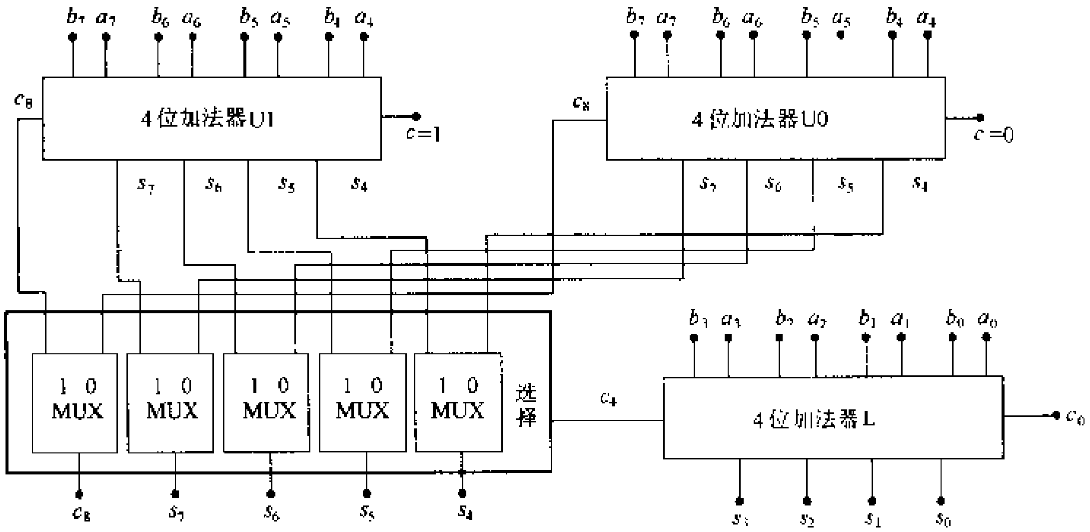


图 12.35　8 位进位选择加法器

这个设计可以使上面部分和下面部分的“和”同时计算，从而加速两个字的相加过程。其代价是它需要增加一个字加法器，一组 MUX 以及相应的互连线。如果速度比起所占面积来更为重要，那么这一设计就十分适宜。进位选择加法器也可以采用多层构成，但硬件成本会相应增加。

12.4.3　进位保留加法器(Carry-Save Adder)

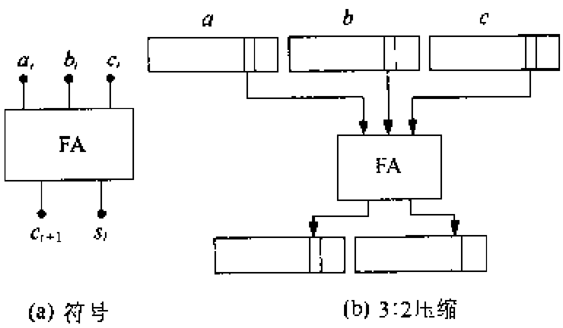


图 12.36　进位保留加法器的基本概念

进位保留加法器所基于的概念是一个全加器实际上有三个输入而产生两个输出，如图 12.36(a)所示。虽然我们通常把第三个输入与进位输入联系起来，但它同样也能作为一个“普通”值。在图 12.36(b)中，FA 用来作为一个 3:2 的压缩电路，即它从三个一位的字开始，把它们相加，然后产生一个两位的输出。可用 n 个单独的加法器来构成一个 n 位的进

位保留加法器,如图 12.37 所示。“进位保留”这个名字来自如下事实,即先保留进位输出字而不是立即用它来计算最终的和。

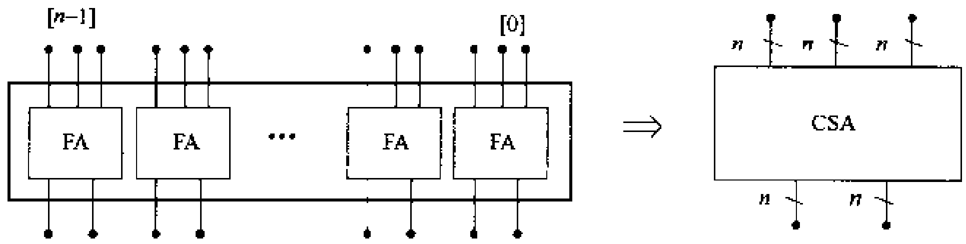


图 12.37 构成一个 n 位的进位保留加法器

进位保留加法器(CSA)在需要相加两个以上的数时非常有用。由于这一设计自动地避免了在进位输出位中的延时,因此一个 CSA 链比起用标准的加法器或者用时钟控制的同步电路周期性地操作要快。

图 12.38 是 7:2 压缩技术的例子。它从 7 个 n 位的字 $a, b, \ldots g$ 开始,用 5 个 CSA 单元把它简化成两个字。如果希望得到最终的和,在这个链的下面可用一个通常的 CPA(进位传播加法器)把这两个值加在一起。

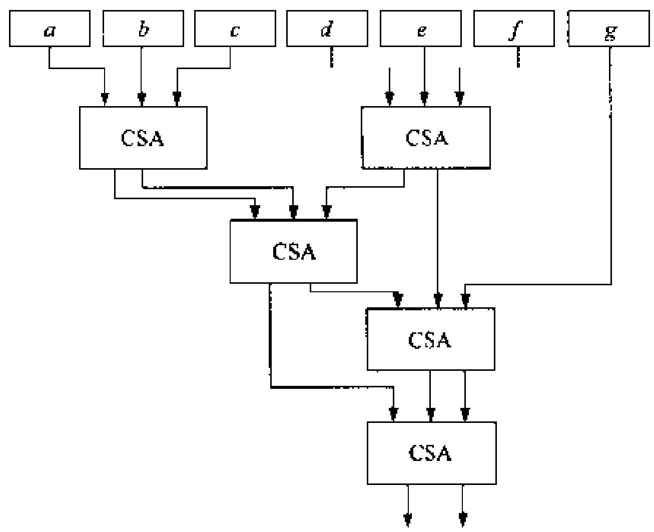


图 12.38 采用进位保留加法器实现 7:2 的压缩

12.5 乘法器

二进制乘法器是以如下的基本操作为基础的:

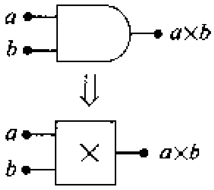
$$0\times0=0\ ,\ 0\times1=0\ ,\ 1\times0=0\ ,\ 1\times1=1 \tag{12.47}$$

如果两个一位 a 和 b 相乘,那么可以看到逻辑“与”操作(AND)产生了同样的结果,如图 12.39 中符号所表述的那样。因此一位的乘法是一个微不足道的操作。

当 n 位字乘时情形就会变得复杂。让我们以字长 $n=4$ 为例来说明这个主要概念。当输入值为 $a=a_3a_2a_1a_0$ 和 $b=b_3b_2b_1b_0$ 时,积 $a\times b$ 为以下 8 位($2n$)的结果:

$$p = p_7p_6p_5p_4p_3p_2p_1p_0 \tag{12.48}$$

如图 12.40 所示。每位 b_i 一位位地去乘被乘数 a。积项从最低有效位 b_0 起与被乘数对齐，而下一项(由 b_1 产生)左移一列。当乘数的每位都已使用时就构成了一个阵列。当将每列(第 i 列)相加并考虑从它右面一列(第 $i-1$ 列)来的进位时，就得到积的每一位(p_i)。积的一个简单表达式如下：



a	b	a×b
0	0	0
0	1	0
1	0	0
1	1	1

图 12.39　一位乘法器

$$p_i = \sum_{i=j+k} a_jb_k + c_{i-1} \tag{12.49}$$

				a_3	a_2	a_1	a_0	被乘数
			×	b_3	b_2	b_1	b_0	乘数
				a_3b_0	a_2b_0	a_1b_0	a_0b_0	
		+	a_3b_1	a_2b_1	a_1b_1	a_0b_1		
	+	a_3b_2	a_2b_2	a_1b_2	a_0b_2			
+	a_3b_3	a_2b_3	a_1b_3	a_0b_3				
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	积

图 12.40　两个 4 位字相乘

式中，当$(i-1)\leqslant 0$ 时，$c_{i-1}=0$。

值得记住的一种特殊情形是乘以 2。对于一个 8 位字的乘数，这相应于输入 $b=00000010$，因此乘以 2 相当于对被乘数进行左移(shl)操作。乘以 $4=2^2$ 即为 $b=00000100$，等等。一般地乘以 2^m 可以通过对存有被乘数的寄存器进行左移 m 位(shl _ m)的操作来实现，如图 12.41 所示。除以 2^k 可以用右移(shr _ k)命令来完成。需要注意是否出现上溢以保证结果是正确的。

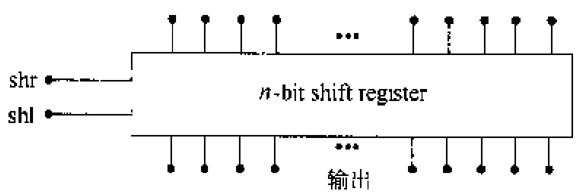


图 12.41　乘以或除以 2 的倍数的移位寄存器

4×4 位乘法器的 Verilog 高层次描述可以写成如下：

```
module mult _ 4(product,a,b);
input[3:0]a,b;
output[7:0]product;
assign #(t _ delay)product = a * b;
endmodule
```

它可用在最初的体系结构模拟中。字长可按需要通过向量说明来调整。然而，关键的延时参数 t _ delay 取决于具体实现，它需要有精确的值以实现一个好的设计。

乘法过程的细节产生了用二进制开关电路计算积的具体技术。对乘法过程的一种解释是 b_i 乘以整个字 a；每一项($a\times b_i$)具有一个十进制的加权因子 2^i。如图 12.42 所示。从第一行 $(a\times b_0)\times 2^0$ 开始并继续进行，每个因子 2^i 代表一次移位。对这个过程的一个简单解释可用图 12.43 所示的乘积寄存器来说明。每一项($a\times b_i$)占据一个移位后的位置，把每一列中的项相加就得到积。

$$
\begin{array}{ccccccccl}
 & & & & a_3 & a_2 & a_1 & a_0 & \text{被乘数} \\
 & & & \times & b_3 & b_2 & b_1 & b_0 & \text{乘数} \\
\hline
 & & & & (a_3 & a_2 & a_1 & a_0) & \times b_0 \\
 & & & (a_3 & a_2 & a_1 & a_0) & \times b_1 & \\
 & & (a_3 & a_2 & a_1 & a_0) & \times b_2 & & \\
+ & (a_3 & a_2 & a_1 & a_0) & \times b_3 & & & \\
\hline
p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p & p_0 & \text{积}
\end{array}
$$

图 12.42　对乘法过程的另一种解释

一个实际的实现基于图 12.44 所示的顺序。寄存器的左边允许并行地装人一个 4 位的字。积的产生是通过连续相加和右移操作来完成的。注意相加得到的最高进位输出位是通过把它们移人寄存器最左边一位来考虑的。对于两个 n 位的字相乘，积的算法可以表示成③

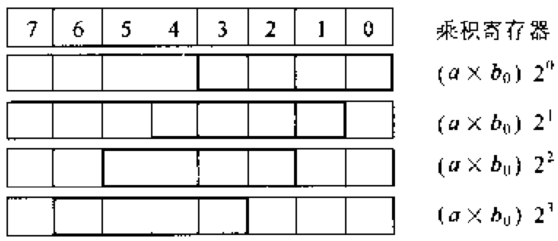


图 12.43　运用乘积寄存器进行乘法运算

$$p_{i+1} = (p_i + a2^n b_i)2^{-1} \qquad (12.50)$$

式中，$p_n = p$ 为最后产生的一位而 $p_0 = 0$。因子 $(p_i + ab_i2^n)$ 为相加操作而 2^{-1} 则代表右移。因子 2^n 乘以 a 是用来在计算结束时补偿由于右移产生的 2^{-n}。这一算法可以用来构成基于寄存器的硬件乘法器电路，如图 12.45 所示。注意乘数位 b_i 用来控制一个 2:1 的多路选择器。如果 $b_i = 0$，把一个 n 位的"0"字送入到加法器；而 $b_i = 1$ 则把被乘数 a 送入到输入端。波茨编码算法可以引人这个电路中，也可以引入几种其他的改进算法。同时还可推导出一种左移算法，得到一种不同的硬件实现。

这一类设计可以用 Verilog 的 assign 语句和移位算子进行编码。为了精确地模拟系统，需要有精确的延时。这些又取决于在库中 VLSI 单元的特性。乘法单元的复杂性反映在 HDL 代码较长以及设计一个高效电路所需要的时间。

12.5.1　阵列乘法器

一个阵列乘法器接受乘数和被乘数并用一个单元阵列以并行的方式计算各个位积 $a_j \cdot b_k$。图 12.46 为一个高层次形式的符号。为了决定这个阵列所需要的特性，把这个高层次形式展开以显示乘法过程的结构，如图 12.47 所示。每个方块要求首先计算位积 $a_j \cdot b_k$，然后把它加入 $i = (j + k)$ 列中的其他项中。这产生了如下的和：

$$p_i = \sum_{i=j+k} a_j b_k + c_{i-1} \qquad (12.51)$$

式中，p_i 指每个积位。这一操作的等效描述可以通过先写出如下的十进制值：

$$A = \sum_{j=0}^{n-1} a_j 2^j \qquad B = \sum_{k=0}^{n-1} b_k 2^k \qquad (12.52)$$

③　见参考资料[2]。

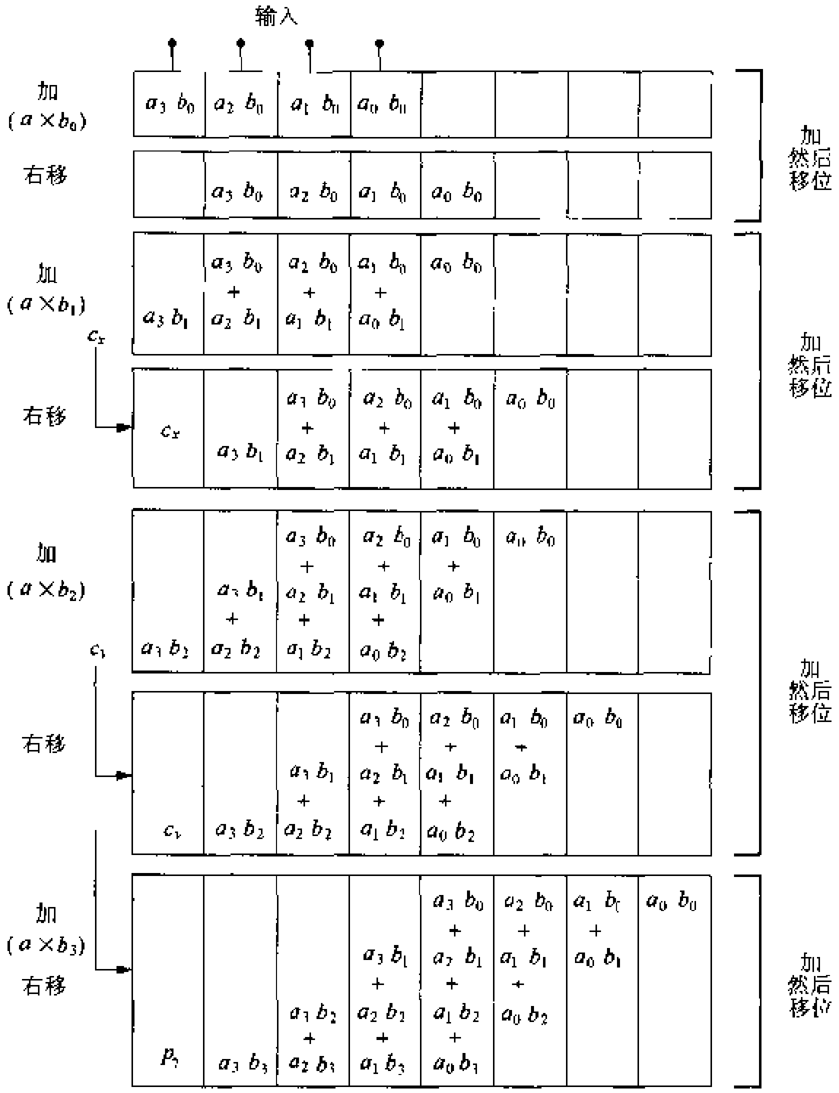


图 12.44　右移操作的乘法顺序

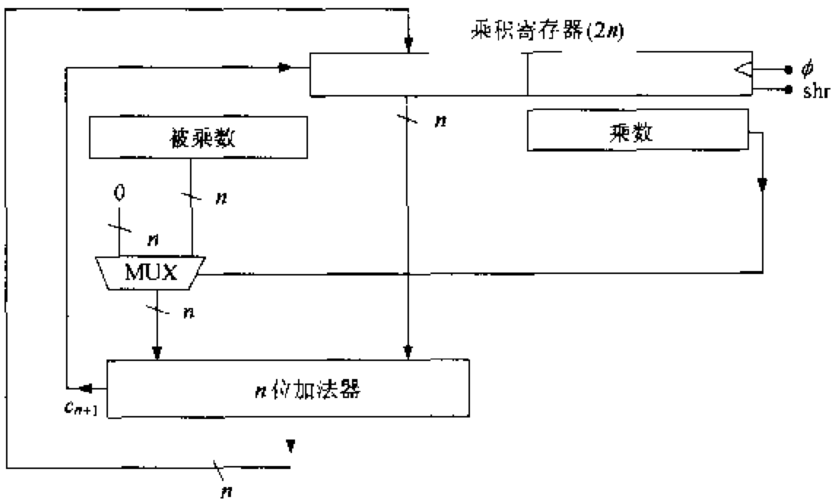


图 12.45　基于寄存器的乘法器电路

然后形成如下的积：

$$
\begin{aligned}
P &= AB \\
&= \left(\sum_{j=0}^{n-1} a_j 2^j\right)\left(\sum_{k=0}^{n-1} b_k 2^k\right) \\
&= \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_j b_k 2^{j+k}
\end{aligned}
\tag{12.53}
$$

由此看到 $a_j \cdot b_k$ 项提供了这一位的值而 2^{j+k} 为加权因子。

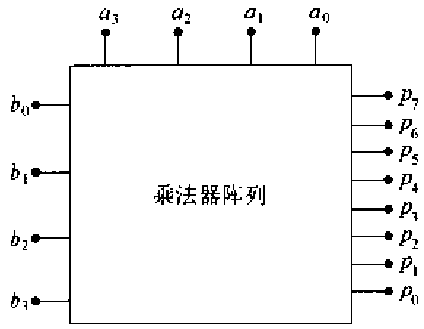


图 12.46　阵列乘法器

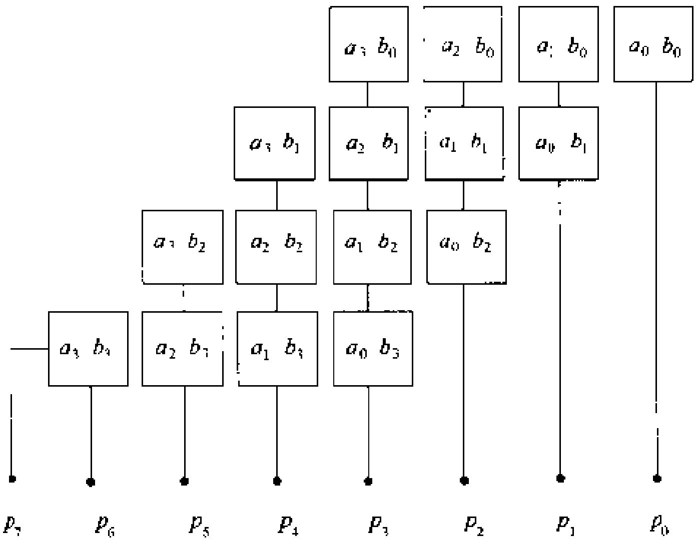


图 12.47　乘法操作顺序的模块化

一个 4×4 阵列的一般结构显示在图 12.48 中。这个结构采用 AND 门来计算位积 $a_j \cdot b_k$。在每列中积的各位是用加法器来形成的。这些加法器组织成进位保留链，这很容易看出，因为进位输出位送到它左边一列的下一行加法器中。阵列乘法器同时接受所有的输入位。计算积位最长的延时取决于加法器的速度。由 p_1 列的进位产生并传播通过 $p_2 \sim p_6$ 的 p_7 进位链显然是一个问题。可以加入一个输入寄存器以使数据流同步，如图 12.49 所示。若有必要也可以用一个输出寄存器。一般地，一个 n 位字的阵列乘法器需要 $n(n-2)$ 个全加器，n 个半加器，n^2 个 AND 门。基于库单元可以用门的数目来估计所要求的面积。

就版图设计而言，如果能使单元安排成总体上比较像矩形形状是很有利的。最初的布局可以使输入位采用规则的互连形式，然后按数据流的次序放置单元本身。这个阵列结构逐渐扩展，它最初部分的样式如图 12.50 所示。实际的布局可做一些调整，以包含互连线及不同的单元尺寸。

12.5.2　其他乘法器

文献中发表了许多种乘法算法及电路。Baugh-Wooley 乘法器是以补码数为基础的，可用于带符号的运算。在这一情形中，把输入数 A 和 B 写成补码形式

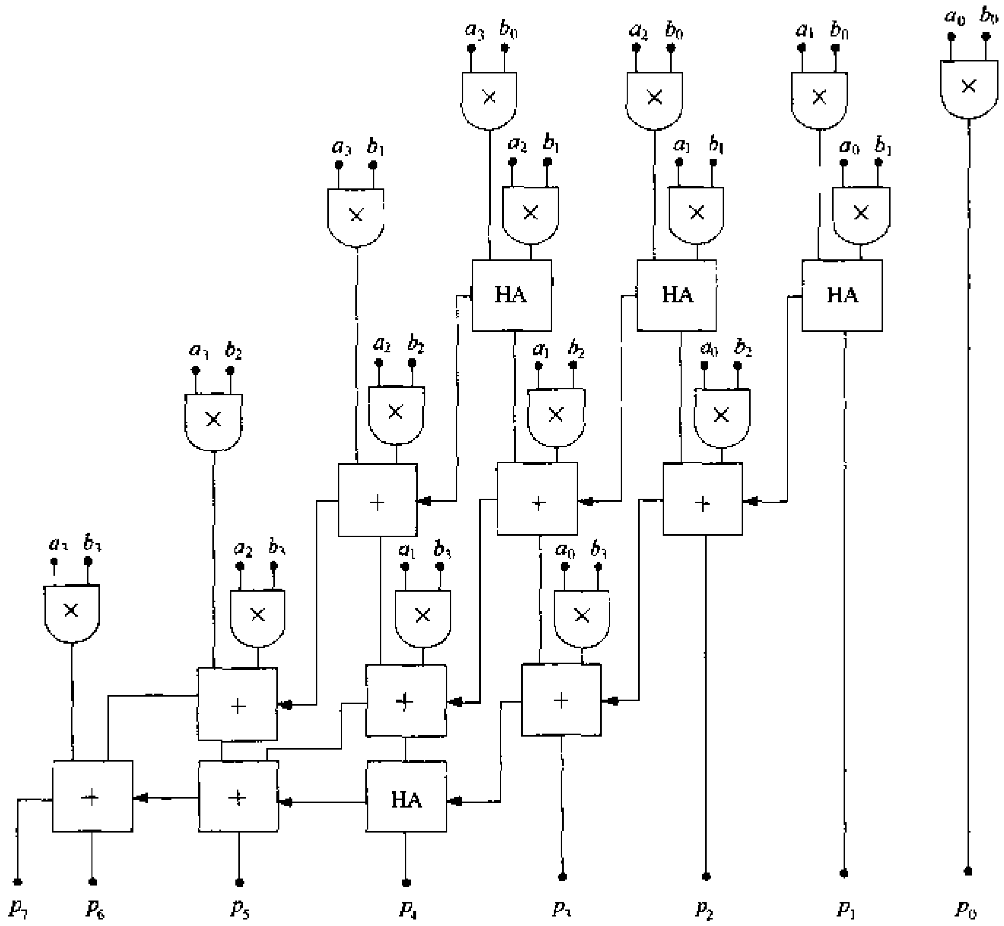


图 12.48 4×4 阵列乘法器详图

$$A = -a_{n-1}2^{n-1} + \sum_{j=0}^{n-2} a_j 2^j \tag{12.54}$$

及

$$B = -b_{n-1}2^{n-1} + \sum_{k=0}^{n-2} b_k 2^k \tag{12.55}$$

于是积为：

$$\begin{aligned} P = {} & a_{n-1}b_{n-1}2^{2(n-1)} + \sum_{j=0}^{n-2}\sum_{k=0}^{n-2} b_k a_j 2^{j+k} + -a_{n-1}\sum_{k=0}^{n-2} b_k 2^{k+n-1} \\ & -b_{n-1}\sum_{j=0}^{n-2} a_j 2^{j+n-1} \end{aligned} \tag{12.56}$$

它可以用加法器和减法器来实现。采用位求反也可以把它转变成只用加法器的实现[1]。

波茨算法与在基本的计算机结构[5]中学习的一样，它考察乘数字 B 并搜索其中的“0”，

因为这些"0"对于求和结果没有任何影响。这可以用来对 B 中的各位进行几位一组的编码以产生控制数字来说明被乘数 A 需要执行什么操作。为了了解这个技术的基础，从补码形式开始：

$$B = -b_{n-1}2^{n-1} + \sum_{k=0}^{n-2} b_k 2^k \quad (12.57)$$

这可以重写成

$$B = \sum_{k=0}^{\frac{n}{2}-1} [b_{2k} + b_{2k-1} - 2b_{2k+1}]2^k = \sum_{k=0}^{\frac{n}{2}-1} E_k 2^k \quad (12.58)$$

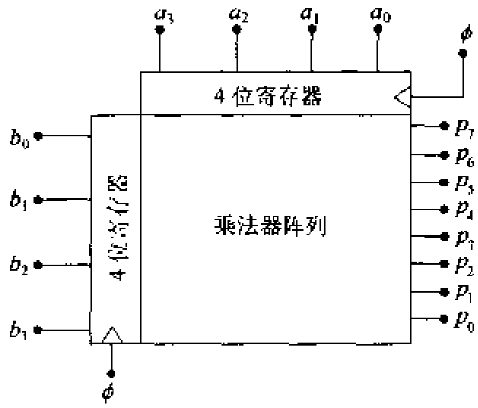


图 12.49　时钟控制的输入寄存器

式中，$b_i = 0$ 及

$$E_k = b_{2k} + b_{2k-1} - 2b_{2k+1} \quad (12.59)$$

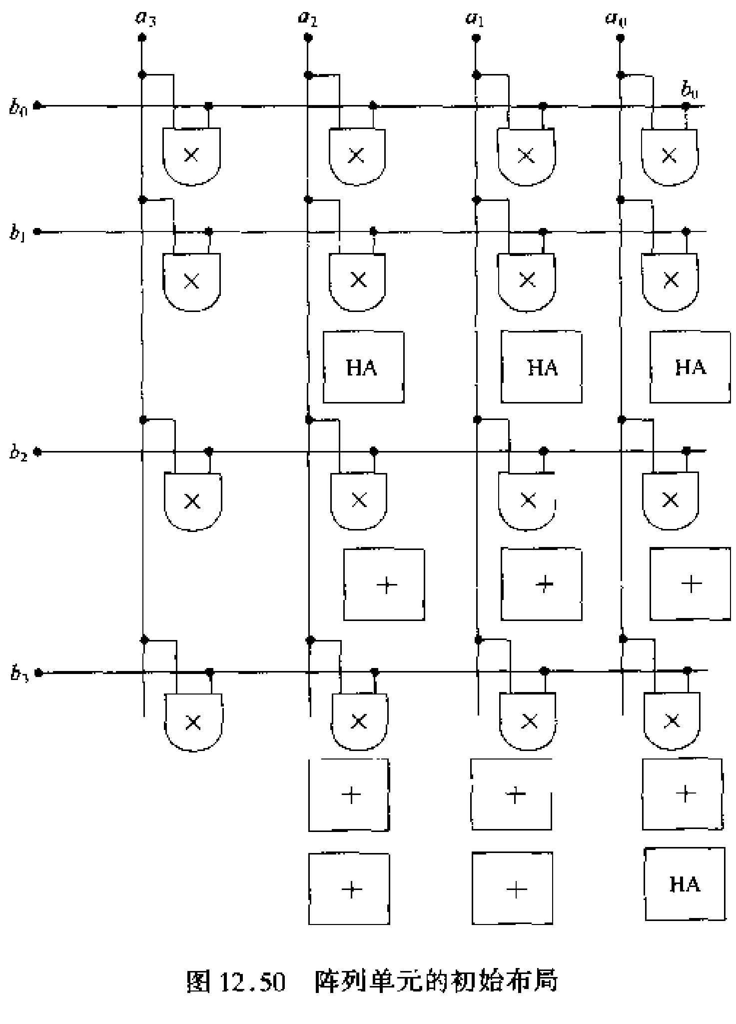


图 12.50　阵列单元的初始布局

为编码数字。因为 b_k 的值为 0 或 1，E_k 可以具有十进制值 +2，+1，0，−1，−2。为了计算积 $A\times B$，把 B 划分成 3 位一组，它们互相重叠一位。例如 8 位字 B = 10**0**11**1**01**0** 可以被分组成

$$\mathbf{100,\ 011,\ 101,\ 100} \quad (12.60)$$

其中重叠的位用粗体字表示。右边最后一个 0 是添加上去的，即 $b_{-1} = 0$。每组产生一个 E_k

值用来决定一个操作。积的计算是通过提供一个双倍字的寄存器，它在每个操作完成之后就存有所产生的和。图 12.51 的表格总结了编码值的含义。对于所示的例子，编码数字为 E = −2，+2，−1，−2。因此 VLSI 电路可以用比较简单的逻辑连同标准的加法单元来构成，这使它对于乘较大的字十分吸引人。另一种称为 Wallace 树的加法电路可以用来改进这个电路，它采用进位保留加法器求和。

b_{2k+1}	b_{2k}	b_{2k-1}	E_k	对求和的影响
0	0	0	0	加 0
0	0	1	+1	加 A
0	1	0	+1	加 A
0	1	1	+2	左移 A，相加
1	0	0	−2	求 A 的补码，左移，相加
1	0	1	−1	加 A 的补码
1	1	0	−1	加 A 的补码
1	1	1	0	加 0

图 12.51 波茨编码的数字操作情况概括

12.6 小结

运算电路采用二进制算法使结构很好地适合 VLSI 的规则版图、单元复用及快速电路的原则。本章中已经考察了与具体实现相关的一些重要问题，但只介绍了最基本的东西。高基数进制的算法，浮点数字以及其他许多研究问题则等待着希望进行更深入研究的感兴趣的读者。

随着微处理器和其他 VLSI 电路不断改进而具有更高水平的性能，运算电路将继续具有关键的重要地位。这为未来辛勤努力的研究者提供了一个迷人的领域。

12.7 参考资料

[1] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[2] James M. Feldman and Charles T. Retter, **Computer Architecture**, McGraw-Hill, New York, 1994.

[3] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2000.

[4] Behrooz Parhami, **Computer Arithmetic**, Oxford University Press, New York, 2000. A comprehensive, in-depth treatment of the subject.

[5] David A. Patterson and John L. Hennessy, **Computer Organization & Design**, 2nd ed., Morgan-Kaufmann Publishers, San Francisco, 1998.

[6] Jan M. Rabaey, **Digital Integrated Circuits**, Prentice Hall, Upper Saddle River, NJ, 1996.

[7] Bruce Shriver and Bennett Smith, **The Anatomy of a High-Performance Microprocessor**, IEEE Computer Society Press, Los Alamitos, CA, 1998.

[8] William Stallings, **Computer Organization and Architecture**, 4th

ed., Prentice Hall, Upper Saddle River, NJ, 1996.

[9] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[10] Neil H.E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, Reading, MA, 1993.

[11] Wayne Wolf, **Modern VLSI Design**, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ, 1998.

12.8 习题

[**12.1**] 采用准 nMOS 设计一个输入为 a 和 b 的半加器。然后用 nmos 和任何其他所需要的基本单元写出 Verilog 的门级描述。

[**12.2**] 考虑 CMOS 双轨 CPL 逻辑系列。

(a) 基于图 12.5(a)的两输入阵列画出一个半加器电路的电路图。

(b) 采用 nmos 基本单元写出一个两输入阵列的 Verilog 模块描述。

(c) 采用图 12.5(b)部分中的两输入阵列模块建立 CPL 全加器的模型。

[**12.3**] 画出用以下每种 CMOS 工艺实现的 4 位 CLA 所需要的 p_i 和 g_i 的电路：

(a) 静态 CMOS;(b)多米诺 CMOS;及(c)TG 逻辑。

[**12.4**] 采用串并联 nFET - pFET 结构构成 CLA $\bar{c}_2$ 位和 $\bar{c}_3$ 位的 CMOS 电路。在每个电路中指出最长的延时路径。

[**12.5**] 参考图 12.20 和 12.21,设计 CLA $\bar{c}_3$ 位和 $\bar{c}_4$ 位的静态镜像电路。

[**12.6**] 考虑图 12.25(a)所示的静态曼彻斯特进位电路。如果 $V_{DD}=3V$, $r=2.5$, $k'_n=150\mu A/V^2$,及 $V_{Tn}=|V_{Tp}|=0.7V$,研究进位传播过程中确定 FET 尺寸的问题。

[**12.7**] 考虑图 12.26 中的动态曼彻斯特进位链。

(a) 画出从 c_0(反相器的输出)至 $\bar{c}_4$ 的进位链的 RC 等效电路。假设每个管子的电阻为 R,每个门的输出节点具有电容 C_{out}。

(b) 这一链在 $\phi=0$ 时预充电而在 ϕ 切换成 1 时进行求值。在求值期间一开始时 $\bar{c}4$ 的值是什么?

(c) 电荷泄漏将如何影响这个进位键的工作?

[**12.8**] 考虑一个 64 位加法器和一个 128 位加法器。每个加法器进位链的平均长度是多少?

[**12.9**] 用 4 位加法器块设计一个 16 位的进位选择加法器。

[**12.10**] 设计一个具有锁存输入的 2×2 阵列乘法器。然后写出设计的 Verilog 描述。

[**12.11**] 考虑图 12.48 中的 4×4 阵列乘法器。能否用它作为模块来构成一个 8×8 阵列乘法器? 如果可以,则详细列出这个问题及所需要的修改。

[**12.12**] 提供一个 8×8 阵列乘法器的基本设计。为构成这个电路需要多少个加法器、全加器,等等?

[**12.13**] 对以下字决定波茨编码数字 E_k:

(a) A=10110011

(b) A=01101101

(c) A=01010010

第 13 章　存储器与可编程逻辑

存储器在现代数字系统中是必不可少的。它们用来短期和长期地存储二进制变量和字。VLSI CMOS 存储器的特点值得研究，因为它们采用单元库设计并且版图具有重复的几何形状。本章讨论半导体存储阵列的设计，且在最后介绍更为一般的可编程逻辑结构。

13.1　静态 RAM

缩略词 **RAM** 代表**随机存取存储器**，这是指一个存储器阵列可以按需要存取任何位（或一组位）。然而在实际中“RAM”的意义已发展为指同时具有读和写能力的存储器以与只读存储（ROM）阵列相区别。

静态随机存取存储器（**SRAM**）单元采用一个简单的双稳态电路来保持一个数据位。一个静态 RAM 单元只要电源一直加在这个电路上，就可以一直保持所存放的数据位。SRAM 具有三种工作模式。当单元处在保持（hold）状态时，位值存放在单元中以备以后使用。在写（write）操作期间，一个逻辑 0 或 1 被送入单元存储。而在读（read）操作期间，所存放的位值被传送至外部电路。

图 13.1 显示了一般的电路形式。一对交互耦合的反相器提供了存储功能，而两个存取管 MAL 和 MAR 用于读和写操作。存取管是由定义工作模式的**字线**信号 WL 控制的。当 $WL=0$时，两个存取 FET 截止使存储单元隔离。这定义了保持状况。为了执行一个读或写操作，使字线上升到 $WL=1$。这就使存取管导通把双轨数据线 bit 和$\overline{\text{bit}}$的存取连到外部电路；这两条数据线常常分别称为 **bit 和 bit 反**。写操作是通过把电压放在 bit 和$\overline{\text{bit}}$线上，作为输入。双轨逻辑帮助提高写的速度。对于读操作，bit 和 bit 反信号线是存储单元的输出，它们被送入决定存储状态的**灵敏放大器**中。在读和写操作之间的区别是单元阵列外部的电路。

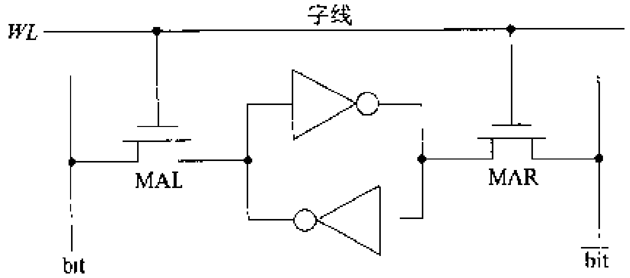
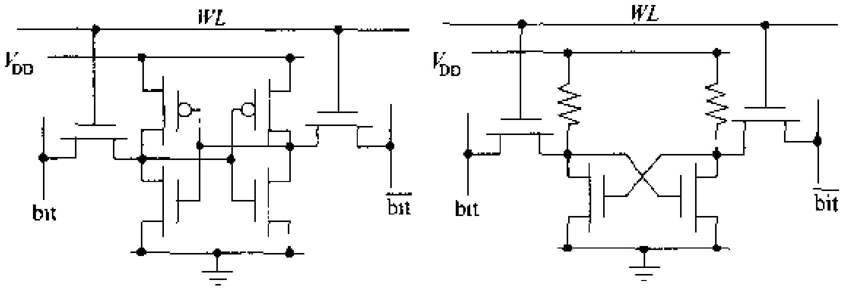


图 13.1　一般的 SRAM 单元

在实际中有两类主要的 CMOS 存储单元。图 13.2 (a)的电路称为 6 管(6T)设计，采用标准的 CMOS 反相器。图 13.2(b)所示的 4 管(4T)单元采用电阻作为 nMOS 电路的负载器件。

电阻是用未掺杂的多晶层制造的，它处在硅(晶体管)层的上面。这样，可使单元面积较小，允许较高的集成密度，但要求工艺增加一层多晶硅及掩模步骤。这两种单元的电气特性相当不同，这是因为 4T 单元采用非常大的无源上拉电阻(通常大约比 1GΩ 还要大)。这里将集中讨论占主要地位的 6T 设计。



(a) GT单元　　(b) 多载电阻负载的4T单元

图 13.2　CMOS SRAM 电路

基本的电路设计问题包括选择管子的宽长比值以保证存储单元可以保持存储状态，仍允许它在写操作期间不需要太长的延时就能改变状态。图 13.3 显示了主要的参数。这里假设是对称设计，即两个存取管子的器件互导都是 β_A，而存储单元本身则采用尺寸分别为 β_n 和 β_p 的 nFET 和 pFET。

保持状态的稳定性取决于交叉耦合反相器单元的工作。反相器的比率(β_n/β_p)确定了每个 NOT 门的中点电压 V_M，后者又确定了反馈特性。这通常是用称为**蝶形图**的曲线来描述的。蝶形图可以这样来获得：在内部的一个节点上施加输入且在另一边画出响应，然后在另一边上完成相同的步骤。这两个图相叠加就得到图 13.4 中的蝴蝶形状。图中标以 SNM 的**静态噪声余量**则是在这两条曲线之间沿 45°斜率的间距，它的单位是伏。这个值说明存储单元对于因电磁耦合信号引起的统称为**噪声**的、不希望有的电压变化所具有的抗扰能力水平。为能稳定存储，需要有一个合理的噪声余量。6T 单元比起电阻负载的 4T 设计来具有更大的 SNM 值，这使它在噪声较大高度密集的情形中更令人感兴趣。① 尽管调整 β_n 和 β_p 的值可以得到不同的蝶形特性，但存储 FET 通常都选择具有尽可能最小的宽长比以使一个 SRAM 阵列的存储密度最大。

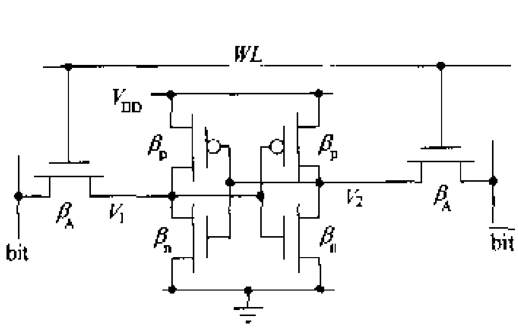


图 13.3　6T SRAM 单元的设计参数

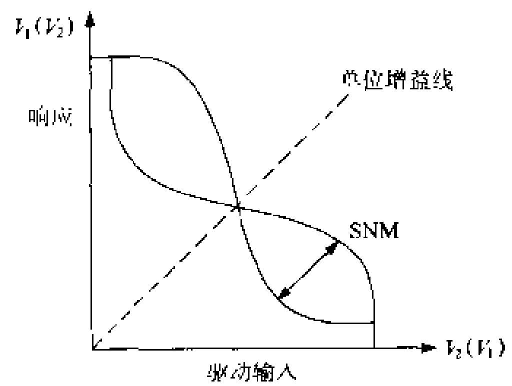
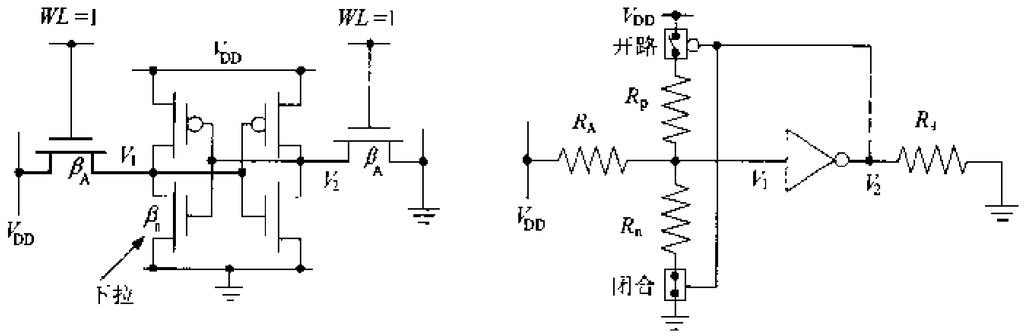


图 13.4　蝶形图

① 电气噪声的问题将在第 14 章有关互连线分析时讨论。

存储单元的写特性可从图 13.5(a)中理解。在这个情形中，把逻辑 1 的位线电压 V_{DD}加到送入存取 FET 的左边位线上，而右边(反信号位线)同时被加上逻辑 0 电压 0 V。最坏情形的状况是最初 $V_1=0$ V 和 $V_2=V_{DD}$，因为无论是位线电压还是反相位线电压都必须改变内部的电压值。重要的设计参数是(β_A/β_n)，对于 6T 单元通常的值在 2 左右。其理由可以从图 13.5(b)所示的电路电阻模型中看出。输入电压 V_{DD}用来把 V_1 提升到逻辑 1 电平。但 nFET 开关(在 R_n 的下面)闭合且把 V_1 拉至 0 V，而且另一个反相器的反馈回路试图保持这个值。选择 $\beta_A>\beta_n$ 意味着 $R_A<R_n$，这使存取 FET 更有效地把 V_1 提升到为改变存放状态所需要的电平。如果单元面积是一个优先考虑的因素，那么(β_A/β_n)的值可以选择接近于 1。注意由于两个 FET 均为 n 型沟道器件，所以它们在版图中设计的尺寸比就简化成宽长比的比率：

$$\frac{(W/L)_{nA}}{(W/L)_n} \tag{13.1}$$



(a) 写 1 操作　　(b) 电阻模型

图 13.5　写 SRAM

在文献中，这常常简单地称为 β 比。

SRAM 单元版图设计时希望同时能使单元面积最小，又使端口的位置允许高密度的阵列。图 13.6 为单元设计的一种方法，它采用相互垂直的金属线 Metal 1 和 Metal 2 来形成电源线(V_{DD}和 V_{SS})及位线(bit)与反相位线(bit-bar)。存储单元则包含在这个单元的中心部分。nFET 的 n+区域被延伸到反相器之外形成存取管，多晶(poly)字线 WL 则垂直走向。如果允许，多晶线可以 45°转弯则有助于缩小面积。

多口 SRAM(Multiple-port SRAM)单元的单元存取有一对以上的位线/反相位线。图 13.7 为一个两端口的单元。字线 WL_1 控制了 bit_1 位线的读/写操作，而 WL_2 则对 bit_2位线提供相同的控制。必须增加逻辑电路避免在这两个端口之间发生冲突。多口存储器可以简化系统布线和版图，因为不同的逻辑部分可以共享一个存储块。然而在系统级必须有一种跟踪存储器内容的方法及一种优先权存取的技术以确保正确的工作。

当一个单元库包含 SRAM 时，用它来设计多个单元的组合以构成大的 SRAM 阵列是很有用的。图 13.8 是一个四单元的组合。两条字线表示为 RW0 及 RW1，它们分别控制上面和下面的一对存储单元。两对字线[X0, Y0] 及[X1, Y1]分别用于左边一对和右边一对存储单元；注意 $X0=\overline{Y0}$及 $X1=\overline{Y1}$。如果多个单元的组合包括在一个库中时，通常都会有支持电路以使容易接口。

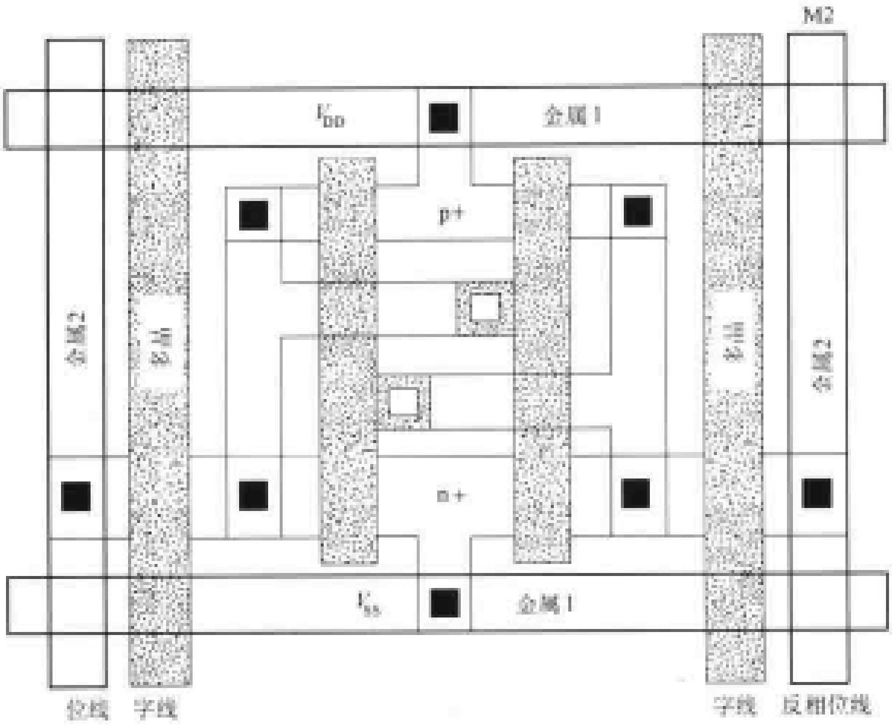


图 13.6　基本 SRAM 单元版图的例子

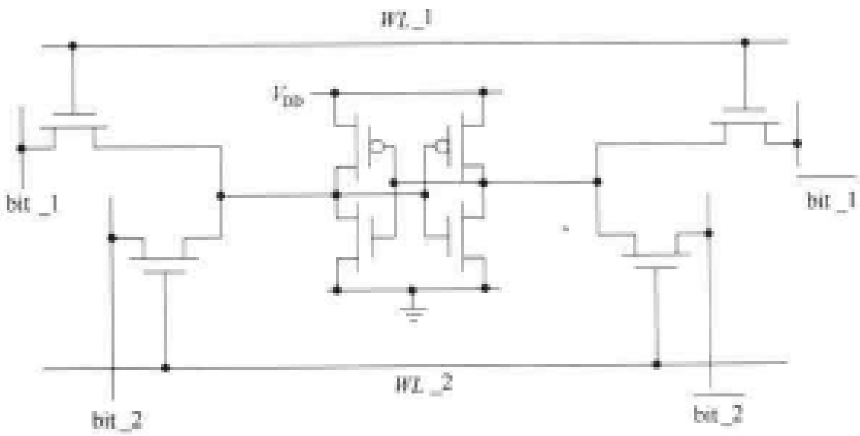


图 13.7　双端口 CMOS SRAM 单元

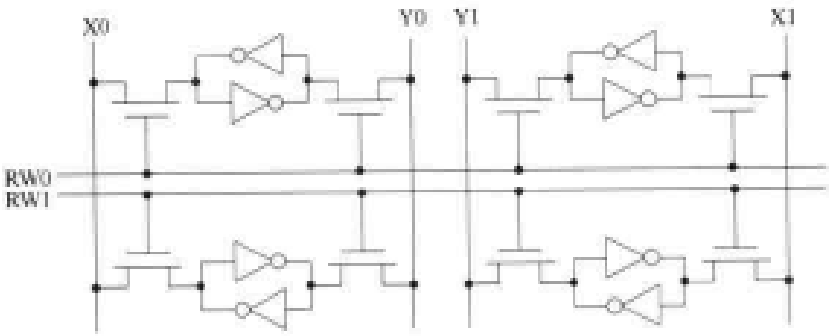


图 13.8　四单元 SRAM 组合

13.2 SRAM阵列

静态RAM阵列是通过复制基本的存储单元并且附加必须的周边电路构成的。其目的是对一个给定的单元版图得到最高的存储密度;在大多数应用中,较短的存取时间也非常重要。

设计一个完整的SRAM提供了对设计层次化的有意义和有用的研究。图13.9为一个SRAM功能单元最高层次的表示形式。在这个层次上一个SRAM含有N个存储位置,每个位置都可以保持一个n位的数据字:

$$D_{n-1}D_{n-2}\dots D_1D_0 \tag{13.2}$$

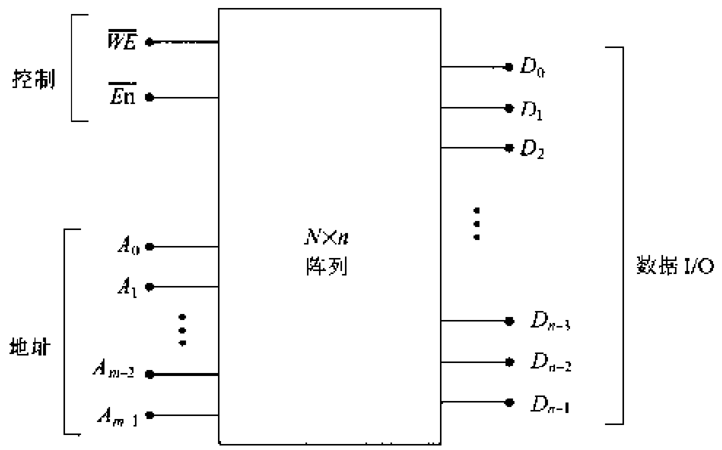


图13.9 SRAM的高层次表示形式

SRAM的容量表示为$N\times n$。一个位置是用一个m位的地址字来说明的:

$$A_{m-1}A_{m-2}\dots A_1A_0 \tag{13.3}$$

其中$N=2^m$能够惟一地选择任何位置。两个控制位包括在图中,这是用来说明读和写操作的。WE是**写使能**信号,图中显示为低电平控制;采用这个表示时,$WE=0$引起一个写操作而$WE=1$表示一个读操作。整个单元是由低电平有效信号En控制的。当$En=1$时,读和写电路不起作用,存储器处在保持状态。需要$En=0$才能启动读/写操作。在芯片层次上,En被重命名为**芯片选择**CS**或芯片使能**CE。

【例13.1】 一个128K×8 SRAM芯片含有128K个8位的字,所以总的储存量为1Mb。为了选择每个8位字的位置,地址线必须具有如下的字宽:

$$\begin{aligned} m &= \log_2(128K) \\ &= 17 \end{aligned} \tag{13.4}$$

Verilog并不提供二维存储器的基本单元,然而在系统级reg数据类型可以写出描述SRAM的语句。一个例子是在以下代码部分中的2KB存储单元sram_1:

```
...
reg [7:0] sram _ 1 [0:2047]
...
```

它用 **reg**[7:0]定义 8 位的字(即一个字节),其标识符为 sram _ 1,地址从 0 至 2047。这可以被修改用于任何字或存储容量。Verilog 高层次描述的简洁性避开了 sram _ 1 单元内部结构的复杂性。为了了解存储器的物理实现,将从组织结构的观点出发逐渐向下并且研究某些电路。这可以用来写出较低层次的 Verilog 模型,它在验证组织结构时是很有用的。在讨论中一个关键点是设计中出现高度的重复性及图形的规则性。

在这里考察的基本组织结构采用两个核心存储单元区,它们共享中间的字线电路,可有许多不同的种类和不同的设计。图 13.10 的方块图为存储阵列中心部分的版图结构。存储单元像铺瓦片那样形成左边和右边核心区域。一个单个的单元显示在这个方块结构图的左边。字线设为沿水平方向走线,而位线和反相位线则按垂直方向布置。一个核心区域的宽度是字宽的倍数。如果采用 8 位的字,那么每个核的宽度为 $k\times 8$,这里 k 是一行中字的数目。图 13.11 是一个核心部分的放大图,它显示这个单元的结构细节。这个结构可以同样适合于本例中的左核和右核。在此电路图中的规则图形可以继续到物理的硅片层次上。单元版图的设计基于找到一种图案,以允许合适的布局布线达到很高的集成密度。

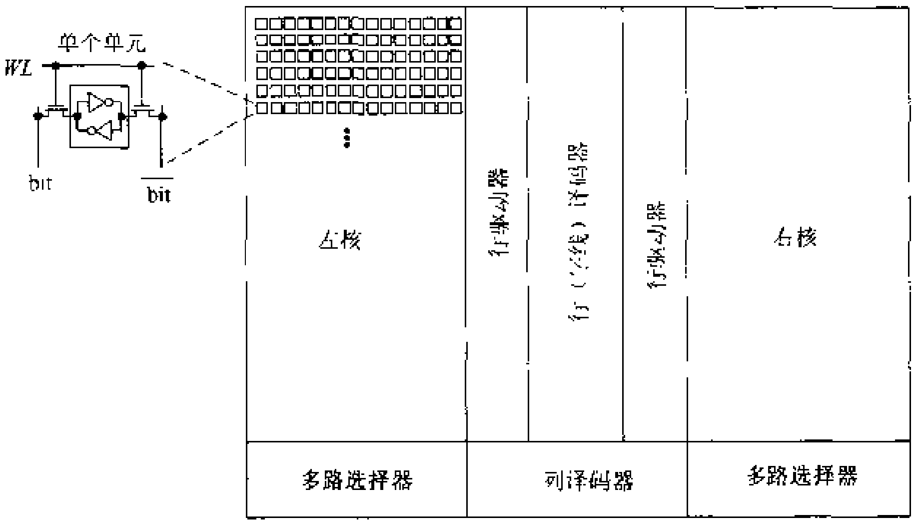


图 13.10　SRAM 块中心部分的结构

一个处于中部位置的高电平控制的行译码器的输出把字线信号送入存储单元中。地址字选中具体一行,然后驱动它至高电平。被选中一行单元的存取管子导通,从而允许读/写操作发生。这一电路的位置允许单个译码器同时用于左边和右边的存储内核。可以把一个库中的静态译码器电路直接例举到这一设计中。行译码器的输出送入到用来驱动阵列字线的行驱动电路。之所以需要驱动器是因为很长的互连线以及连至每条字线的存取管都有很大的电容负载。图 13.12 为一个基本的行驱动器设计。译码器的输出表示为 Dec _ out。第一个 pFET 作为一个上拉器件,而第二个 pFET 则通过围绕 NAND2 门的反馈来加强这一输出。一个尺寸设计好的反相器链用来提供对字线的驱动能力。

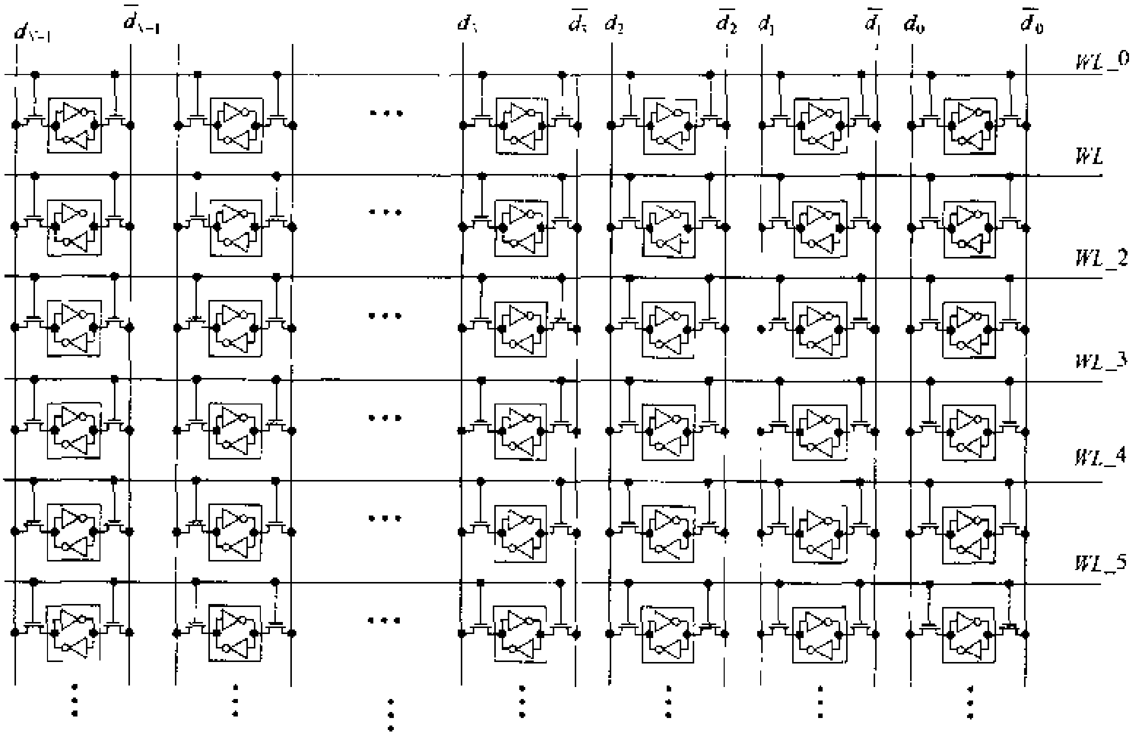


图 13.11 核心区域的单元排布

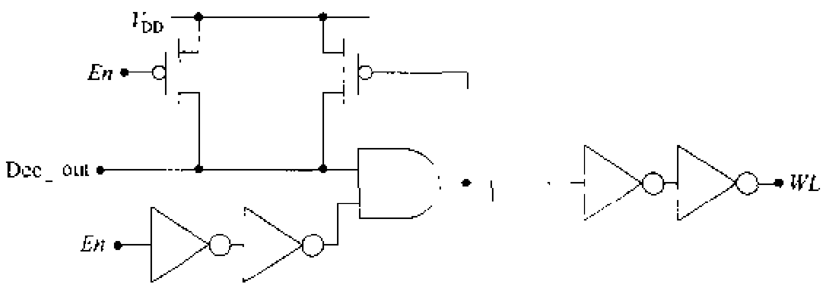


图 13.12 行驱动器电路

从图 13.11 的阵列中可以看到，单元的输入/输出数据位线及反相数据位线形成存储矩阵的列。可以看出，无论读还是写操作数据流都是垂直方向的。一旦一条字线被行译码器驱动至高电平，在该行中的每个单元都是可以存取的。为在这一行中选择一个特定的 k 位字，必须增加一组列译码电路，它们在矩阵中选择特定的一组 k 列。图 13.10 所示的多路选择器 MUX 部分由列译码器控制以把所选择的一组(位，反相位位)信号传送出去。对于 8 位字的情形，列选择电路的整体结构显示在图 13.13 中。每个 MUX/DeMUX 块被连到每个字的合适的数据线上；例如每个字的位线 bit _ 0 和$\overline{\text{bit}}$_ 0 线被连至 Bit0 MUX/DeMUX 块上。一个读操作要求单元的输出，所以这些电路的作用是多路选择器。对于写操作，必须用 DeMUX 模式把数据字送入合适的列中。列驱动器把信号送入 MUX 中；图 13.14 为一个简单的采用反馈的逻辑 1 驱动器设计。当 In = 1 时，输出为 0 使上拉 pFET 导通；pFET 的连接有利于保持高的输入电压。

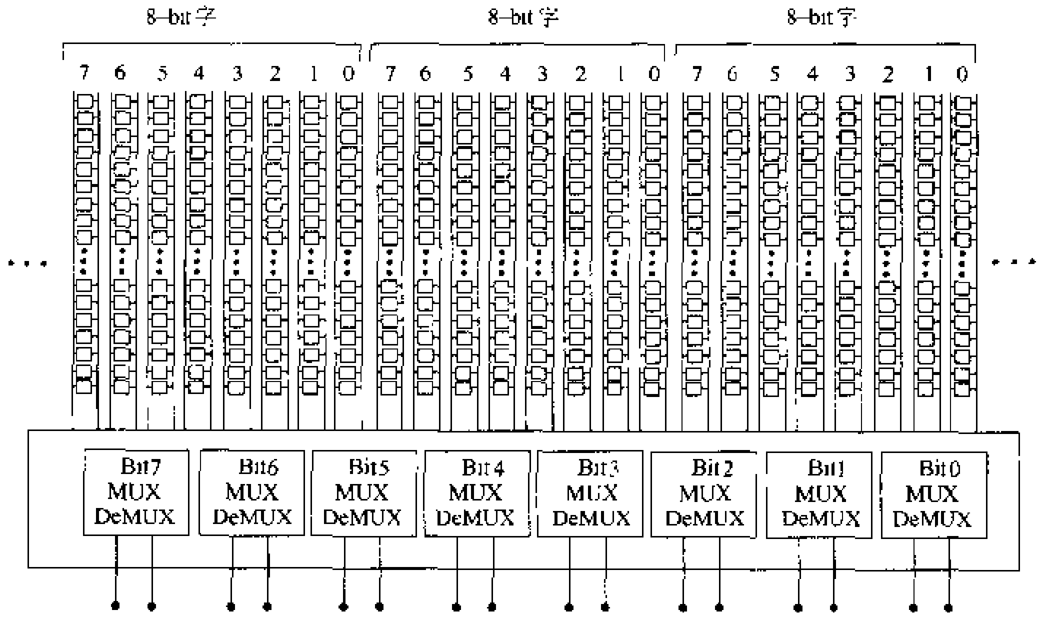


图 13.13　8 位字的列 MUX/DeMUX 电路

为了清楚说明寻址技术，让我们考察最简单的情形，即 m 位地址字 $A = A_{m-1} \ldots A_2 A_1 A_0$ 划分成 x 行一组和 y 列一组，使 $x + y = m$。在图 13.15 的方块图中，地址被送入一个地址锁存器中使之稳定下来。图 13.16 为一个地址锁存电路。它含有使能信号 E 控制的一个基本的 D 型锁存器，使能信号 E 由 En 和其他控制信号产生，用来使系统同步。这一电路在 E 从 0 过渡到 1 时锁存输入地址位 Add_in。输出 A 和 $\bar{A}$ 划分成列和行两部分并且作为译码器电路的输入。这可存取在单元矩阵中的任何组字。

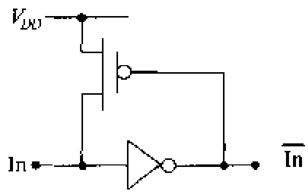


图 13.14　逻辑 1 的列驱动器

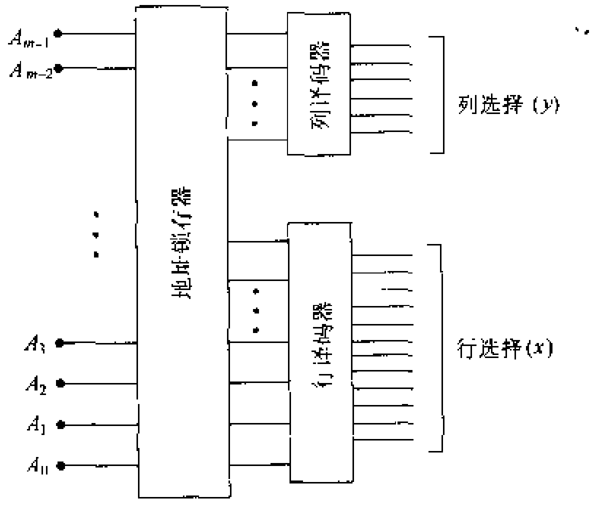


图 13.15　基本的寻址技术

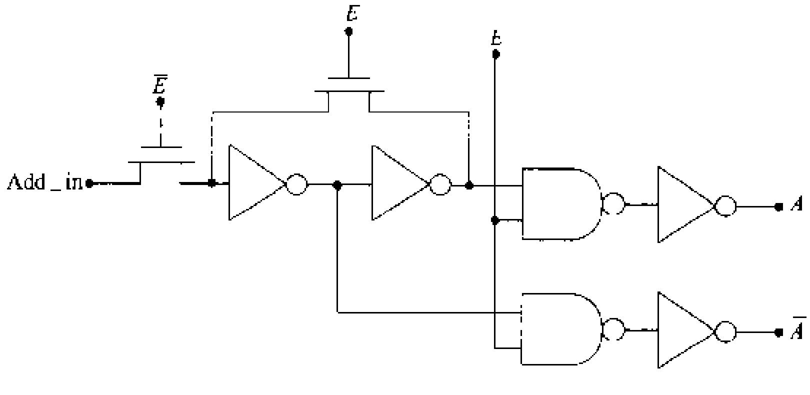


图 13.16 地址锁存电路

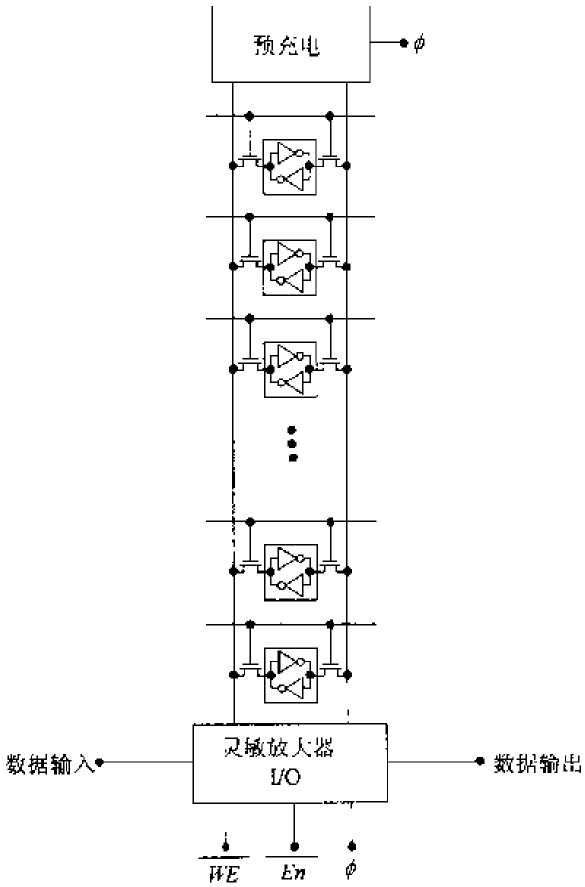


图 13.17 其中一列的预充电和 I/O 电路

【例 13.2】 128 K×8 SRAM 芯片要求 17 位的地址字。如果采用双核结构并且每条字线一个字，那么需要 64 K 字线。若把每条字线的控制范围扩展到 64 位(=8 个字)，那么字线的数目减少到 8 K。于是 17 位的地址 $A_{16}\cdots A_0$ 可被划分成 4 位的列地址组 $A_{16}\ A_{15}\ A_{14}\ A_{13}$ 及 13 位的行地址 $A_{12}\cdots A_0$。其他大小的阵列也可按比例划分地址字。

尽管静态库电路可以构成图 13.10 所示的整个 SRAM 电路，但动态电路通过对具有大电容的 I/O(输入/输出)位线和反相位线进行预充电而加快了读操作。图 13.17 为其中一列的方块图。顶部的预充电电路由时钟信号 ϕ 控制，它用来使操作和数据流同步。读和写操作表示在该列的底部。比较细节的内容显示在图 13.18 的扩展图中。在 $\phi=0$ 的读操作期间预充电电路工作；在这期间，每条数据线上的电压提升至 V_{DD}。当时钟变为 $\phi=1$ 时，则开始求值。在这期间一列的位线和反相位线送入差分"灵敏"放大器，由它确定被存放位的值。该图也显示了列 MUX 电路。每个字由一个控制信号选择；Col _ 0 用来作为该图中的一个例子。当 Col _ 0=1 时，nFET 起作用，整个一组位线和反相位线连至读/写电路方块。对每一组字采用单独的列选择信号。

读/写电路完成以下几个功能：

- 在写操作期间把数据流送入阵列中，或在读操作期间把数据流从阵列中取出；

- 把读和写电路连到每列的位线和反相位线上；
- 提供放大器在读操作期间检测并且放大输出。

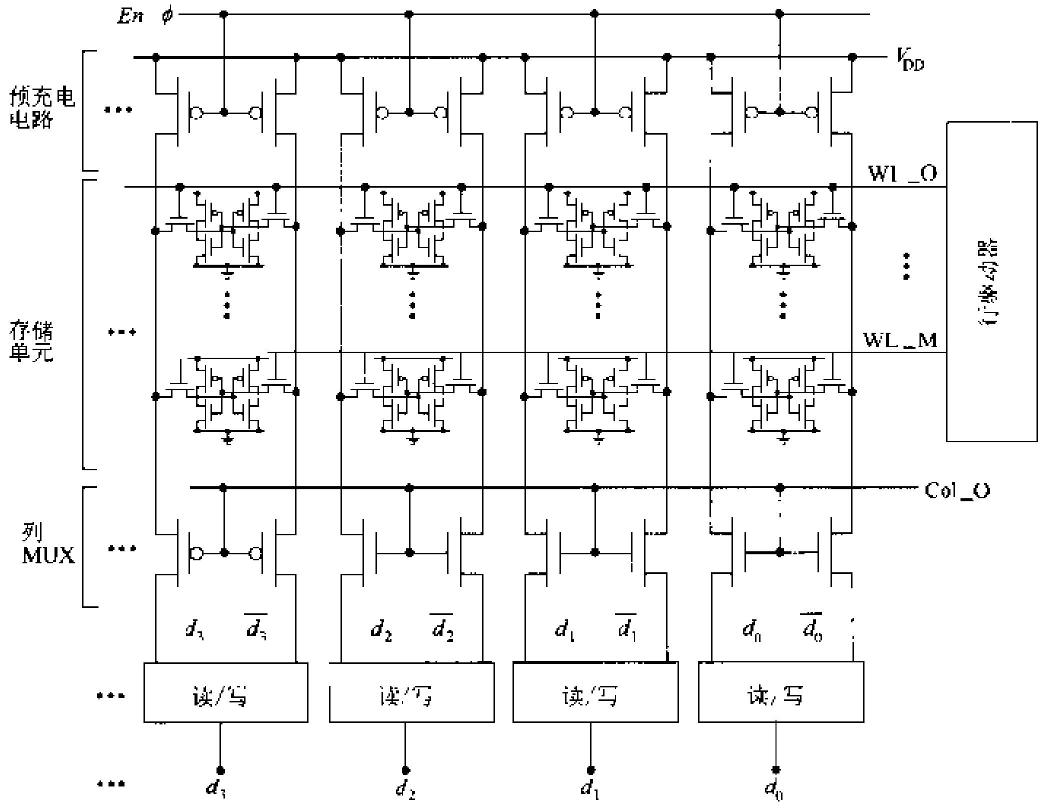


图 13.18 列电路的细节图

对于一个 8 位字的设计，写电路的一个例子显示在图 13.19 中。输入位 $d_7, d_6, \cdots d_1, d_0$ 被反相及缓冲，提供互补的一对信号 $(d_i, \overline{d_i})$。当写使能控制位 $WE=1$ 时，nFET 的作用像一组闭合的开关，它们把一对数据连到位线和反相位线上。每一对数据送入各组 8 位字的合适位置上。列多路选择器电路（图中没有明确表示出来）决定哪一列接受输入字。

在读操作期间要求附加的电路检测所存放位的值。图 13.20 所示的一位电路方块图基于具有 + 和 − 输入的差分放大器（它表示成三角形符号）。每个输出位需要完全相同的电路。差分放大器的输出取决于输入电压 v^+ 和 v^- 之间的电压差

$$v_d = (v^+ - v^-) \tag{13.5}$$

放大器的输出电压为

$$v_{out} = Av_d = A(v^+ - v^-) \tag{13.6}$$

式中，$A>1$ 是放大器的电压增益。当用在 SRAM 中时，差分放大器的输入来自存储单元的位信号和反相位信号。图中所示的电路采用两层灵敏放大技术。第一层含有一对差分放大器，它们被送入相位相反的两个输入。它们的输出一起送到另一个差分放大器，后者的输出送至一个数据锁存器。为使这些与动态的列预充电电路相匹配，这些灵敏放大器本身也是由时钟信号 ϕ 控制的。

图 13.21(a)为差分放大器晶体管级的细节。这是基于两个接受互补输入 d 和 $\overline{d}$ 的输入

nFET Mn1 和 Mn2 的标准设计。pFET Mp1 和 Mp2 作为**有源负载**器件，它们的作用如同非线性的上拉电阻。由位电压和反相位电压形成的差分信号($d-\bar{d}$)控制流过 nFET 的电流 I_{D1}和 I_{D2}。当与 d 相关的电压较大时，I_{D1}增加；类似地，增加 $\bar{d}$ 电压将增大 I_{D2}。总的电流由流经时钟控制 nFET Mn 的电流所限制：

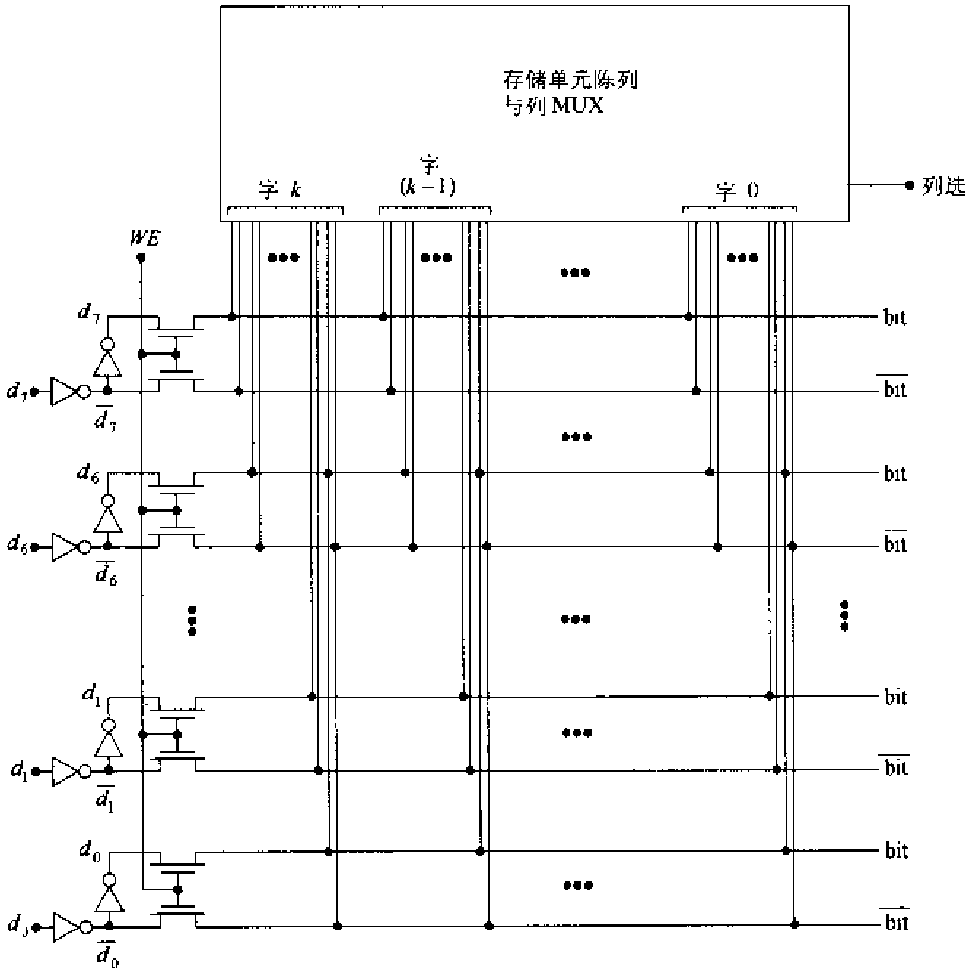


图 13.19 写电路的例子

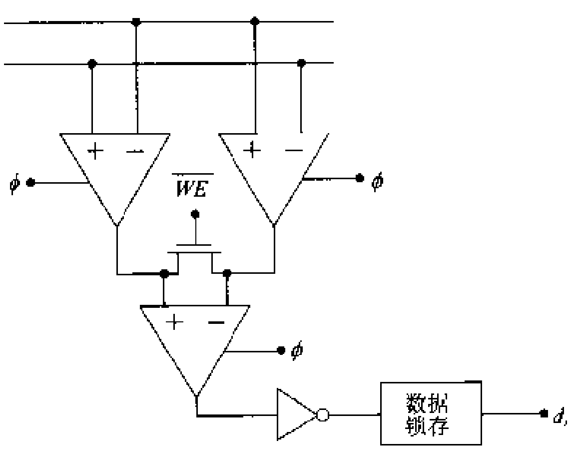
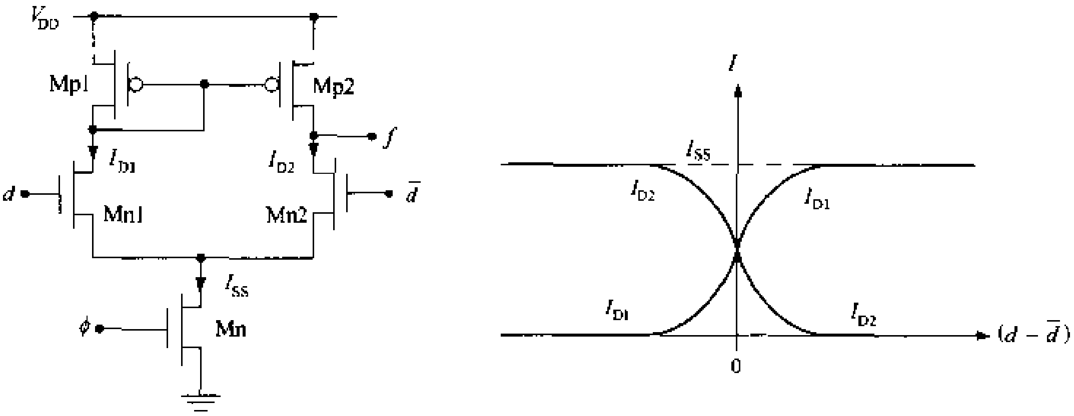


图 13.20 读操作时灵敏放大技术的例子

$$I_{SS} = I_{D1} + I_{D2} \tag{13.7}$$

在本设计中上式在 $\phi = 1$ 时成立。对此电路进行分析得到图 13.21(b)所示的电流特性，它画出作为$(d - \bar{d})$函数的电流。



(a) 电路图 (b) 电流与差分输入的关系

图 13.21 单端输出的差分放大器

当输入相等即 $d = \bar{d}$ 时，$(d - \bar{d}) = 0$，于是 $I_{D1} = I_{D2}$。在 SRAM 读操作期间，一个电压将高于另一个电压。如果 $d > \bar{d}$，则 $I_{D1} > I_{D2}$，而当输入值 $d < \bar{d}$ 时有 $I_{D2} > I_{D1}$。这个电流差被转换成一个低的或高的输出电压。在电路级，设计问题涉及选择晶体管的宽长比，后者又确定了小信号增益。

图 13.22 为第一层的双放大器电路图。它以交叉驱动的方式把两个单个的放大器组合在一起以提高检测电路的灵敏度。一个高的灵敏度意味着读操作将需要较少的时间，这可实现快速 RAM 阵列的设计。这类电路也用于通信应用的高速硅芯片接收电路中。电路的均衡特性使它在减少噪声以及制造工艺变化的影响方面十分吸引人。

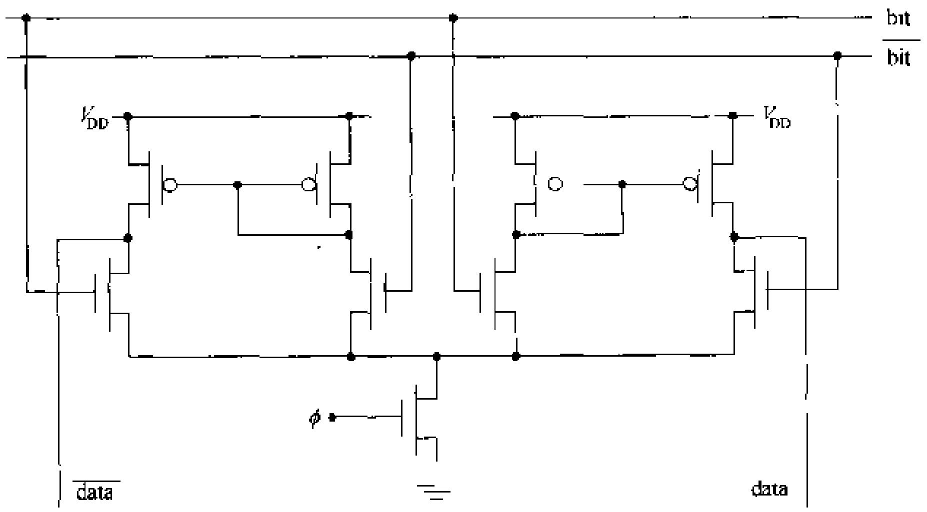


图 13.22 灵敏放大电路的双放大器技术

13.3 动态 RAM

动态 RAM(DRAM)单元要比 SRAM 单元小得多,能够构成较高密度的存储阵列。每位成本的减少使它们在要求大容量读写存储器,例如微计算机系统主存储器的应用中十分吸引人。DRAM 比 SRAM 慢,并且需要较多的外围电路。在电路级,它们结构简单但设计可能较难,特别是当速度是一个问题的时候。

DRAM 存储单元和系统的设计是一门高度专业化的学问,它只能在物理层上进行工作才能完成。然而在 VLSI 系统级,存储器只看成二进制数据的一个存储单元。当 DRAM 存储单元用于 VLSI 设计时,它通常由专门设计组设计好的库实体中例举出来。[②] 考虑到这一观点,对 DRAM 的讨论将局限于理解最基本的东西以了解它的操作和综合设计考虑。

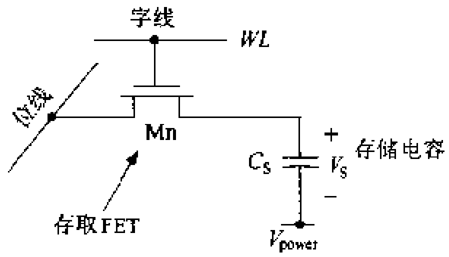


图 13.23 IT DRAM单元

图 13.23 为单管(1T)DRAM 单元。它含有一个存取 nFET Mn 和一个存储电容 C_s。这个单元由字线信号 WL 控制,单根位线把 I/O 路径连到单元中。电容的下端连至两条供电线中的一条,图中它表示成 V_{power},可以是 V_{DD}或者 V_{SS}。存储原理是基于电荷暂时驻留在电容上的概念。电容两端的电压 V_s 相应于存储一个电荷 Q_s:

$$Q_s = C_s V_s \tag{13.8}$$

当 $V_s=0$ 时 $Q_s=0$,此时电荷状态就是逻辑 0。相反,一个大的 V_s 值对应于一个大的 Q_s,而这被定义为逻辑 1 的电荷状态。

图 13.24(a)为 $V_{power}=V_{SS}=0V$ 时的写操作情形。把 V_{DD}加到 nFET 的栅极使存取管导通从而允许访问这个存储电容。输入数据电压 V_d 控制了流进或流出 C_s 的电流。逻辑 0 的数据电压 $V_d=0$ V 使电容两端的电压 $V_s=0$ V,这相应于电荷状态 $Q_s=0$。如果我们应用一个逻辑 1 的数据电压 $V_d=V_{DD}$=电源电压,那么在栅上的电压使被传送的信号减少一个 nFET 的阈值电压。能够传送到电容上的最大电压为

$$V_s = V_{max} = V_{DD} - V_{Tn} \tag{13.9}$$

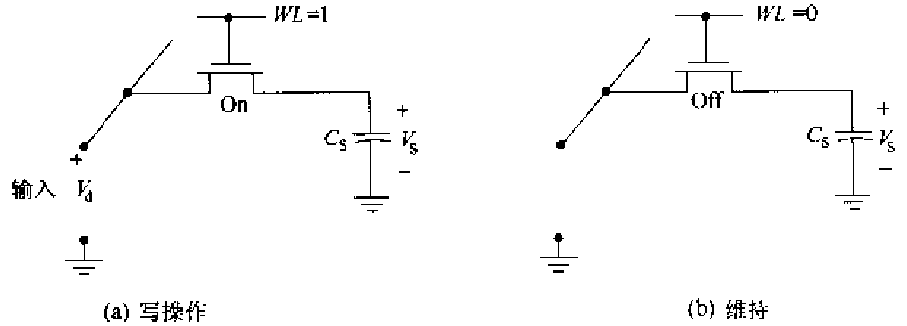


图 13.24 DRAM 单元的写和保持操作

② SRAM 也常是这样考虑的。

这相当于存放最大的电荷

$$Q_{max} = C_s(V_{DD} - V_{Tn}) \tag{13.10}$$

保持状态的实现是通过使字线信号 $WL = 0$ 而使存取管截止。这显示在图 13.24(b)中。

单元的动态特点显示在数据保持期间。正如在第 9 章中讨论的那样,当一个 MOSFET 被偏置为 $V_G < V_T$ 而进入截止状态时仍然允许小的漏电电流。DRAM 电路的问题显示在图 13.25 中。在存储电容上的逻辑 1 电压 $V_s = V_{max}$ 产生了电动力使漏电流 I_L 从 C_s 中流出。这可以用下式来描述:

$$I_L = -\left(\frac{dQ_s}{dt}\right) \tag{13.11}$$

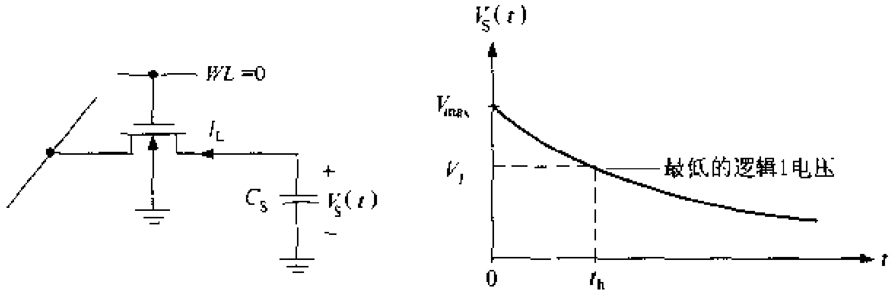


图 13.25 DRAM 单元中的电荷泄漏

上式表明电流从电容中移出电荷。利用 Q_s 的方程(13.8)得到如下有关电容的关系:

$$I_L = -C_s\left(\frac{dV_s}{dt}\right) \tag{13.12}$$

所以 V_s 也下降。假设 V_s 最初的电压为 V_{max},那么电压衰减的情况显示在图 13.25 中。图中最低的逻辑 1 电压表示为 V_1。保持时间 t_h 定义为单元维持一个被认为是逻辑 1 的足够大的电压的最长时间;保持时间在文献中也被称为数据保留(retention)时间。一般地,I_L 是电压的函数,因此求出 $V_s(t)$ 需要解一个非线性方程。在估计 t_h 时可以假设 I_L 是常数而写出:

$$I_L \approx -C_s\left(\frac{\Delta V_s}{\Delta t}\right) \tag{13.13}$$

这里 ΔV_s 和 Δt 代表变量的变化量,重新整理得到保持时间的方程:

$$t_h = |\Delta t| \approx \left(\frac{C_s}{I_L}\right)(\Delta V_s) \tag{13.14}$$

它可以作为初步估计。这表明采用大的电容并使漏电流最小,可以增加保持时间。作为一个例子,如果 $I_L = 1\text{nA}$, $C_s = 50\text{fF}$,而$(\Delta V_s) = 1\text{ V}$,那么保持时间就是:

$$t_h = \left(\frac{50\times10^{-15}}{1\times10^{-9}}\right)(1) = 0.5\mu\text{s} \tag{13.15}$$

这显示了 DRAM 单元的保持时间很短,并且清楚地说明该电路采用形容词“动态”是合理的。

存储单元只要电源存在就应当能保持住数据。为了克服电荷泄漏问题,DRAM 阵列采用了**刷新操作**,即每个单元的数据被周期地读出、放大然后重写。这一过程列在图 13.26 中。这一刷新周期必须针对阵列中的每个单元进行,最低的刷新频率约为:

$$f_{\text{refresh}} \approx \frac{1}{2t_h} \tag{13.16}$$

刷新电路包括在单元阵列周围的附加逻辑中。刷新周期设计成“暗中”进行，因此对于用户而言是看不到的。

读操作显示在图 13.27 中。在读期间电容上的电压 V_s 把电荷从 C_s 移向位线电容 C_{bit}，这就产生了电荷分享的情形。C_{bit}包括位线电容以及其他寄生电容，如灵敏放大器的输出电容。电容上最初的电荷为：

$$Q_s = C_s V_s \tag{13.17}$$

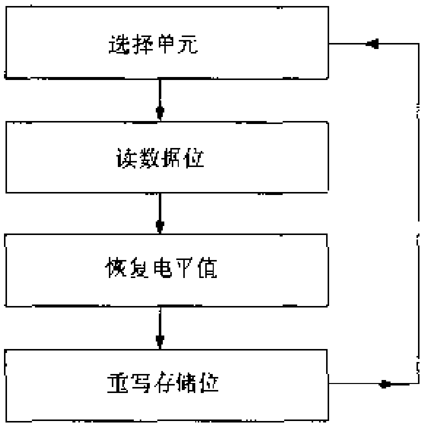


图 13.26　刷新操作

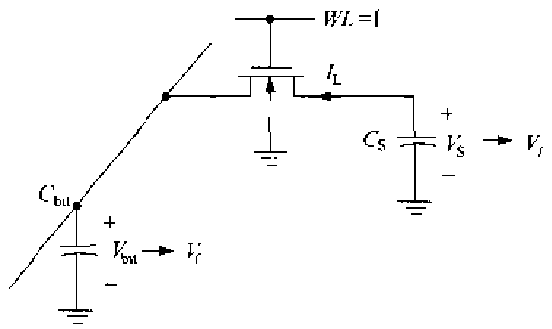


图 13.27　DRAM 单元的读操作

式中对逻辑 0，$V_s=0$ V；而对逻辑 1，$V_s>0$。从 C_s 到 C_{bit}的电流一直继续到最终电压相等即 $V_f=V_{bit}=V_s$ 时为止。电荷重新分布为：

$$Q_s = C_s V_f + C_{bit} V_f \tag{13.18}$$

根据电荷守恒 Q_s 的初值和终值应当相等，所以

$$V_f = \left(\frac{C_s}{C_s + C_{bit}}\right) V_s \tag{13.19}$$

这表明对存储的逻辑 1，$V_f < V_s$。实际上，V_f 通常减少到几百毫伏，所以灵敏放大器的设计是一个关键因素。

【例 13.3】 假设有一个 DRAM 单元，它的 $C_s=50$ fF 而位线电容 $C_{bit}=8C_s$。假设在存储电容上电压 V_s 的最大值 $V_s=V_{max}=2.5$ V，在读逻辑 1 操作期间的最终电压为

$$V_f = \left(\frac{1}{9}\right)(2.5) = 278 \text{ mV} \tag{13.20}$$

由于存储的逻辑 0 使 $V_f=0$ V，所以灵敏放大器必须区分 0 V 和 0.28 V 以确定所存位的值。

13.3.1　DRAM 单元的物理设计

现代 DRAM 芯片由于采用新的电容结构，已经超过 1 Gb 的容量密度，这些新的电容结构由于先进的半导体工艺技术而成为可能。1 T 存储单元是由一个晶体管和一个存储电容构成的。高密度的阵列是通过把单个单元的面积 A_{cell}减到尽可能最小的尺寸来实现的。用于寻址、刷新和其他操作的外围电路必须加入以使芯片能够工作，它们能够很容易地占去整个芯片面积的 30%以上。

在标准的 MOS 工艺中，nFET 必定在硅圆片上；因为亚微米的线宽是很普遍的，所以 FET 的面积很小。减少整个单元的面积通常涉及存储电容的设计。C_s 的值必须大约为 40 fF 或更大。平行板电容公式表明需要的极板面积 A_p 为

$$A_p = C_s\left(\frac{t_{ins}}{\varepsilon_{ins}}\right) \tag{13.21}$$

式中 t_{ins}和 ε_{ins}分别为绝缘层的厚度和介电常数。假设二氧化硅层的厚度为 50Å，这意味着极板面积

$$A_p = (40\times10^{-15})\left(\frac{50\times10^{-8}}{3.45\times10^{-13}}\right) = 5.8\times10^{-8}\ \text{cm}^2 \tag{13.22}$$

即 5.8 μm^2。这个面积太大了，不能在大的阵列中运用。例如一个 64 Mb 的 DRAM 通常要求单元尺寸约为 1.25 μm^2 以满足芯片要求。已经进行许多有关存储电容结构的研究以加大极板的面积，而不增加单元的表面积 A_{cell}(也称为 footprint 尺寸)。有两种主要的结构被采用：沟槽电容和堆叠电容。

采用沟槽电容的存储单元显示在图 13.28 中。这个电容的形成是用反应离子刻蚀(RIE)工艺在硅上形成一个深沟。沟壁被氧化以形成一玻璃绝缘层，运用掺杂多晶硅充满沟槽作为上极板。下极板是沿整个侧壁区域进行 n^+ 注入来形成的。由于利用沟槽的侧壁所以增加了极板的面积 A_p 而不增加单元的表面积 A_{cell}。

堆叠电容的设计是把一个多晶硅结构放在存取管的顶部如图 13.29 所示。可以形成先进的三维结构作为电容的上下极板。极板的面积 A_p 取决于由多晶极板形成的表面几何面积。文献中已经发表许多有趣的堆叠电容的设计。此外还开发了表面沟(corrugations)和"突起"(bumps)以进一步加大 C_s 的值。有兴趣的读者可参考资料[8]以了解这一方面的概况。

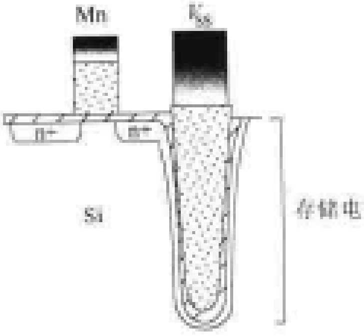


图 13.28　采用沟槽电容的 DRAM 单元

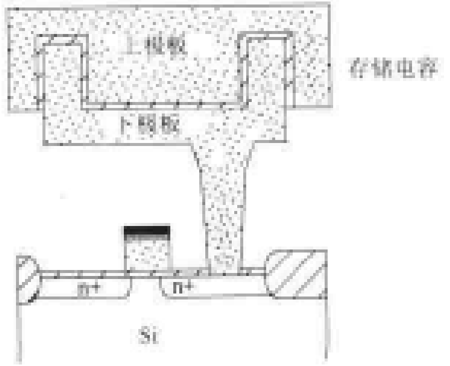


图 13.29　堆叠电容结构

13.3.2 分割字线结构

随着 RAM 容量的增加，已经开发新的版图设计方法以减少由长互连线引起的寄生电阻和电容的影响。采用多个储存内核可以把总的存储面积划分成几个单独的单元块；每块都定义一个特定的地址范围。这些储存块还可以进一步划分成**子块，子块的子块**，等等，直到单元阵列的大小可以被存取而不会有过长的延时。图 13.30 说明了这个概念。分隔存储阵列的优点是字线的布线可被分隔成多条路径。这一**分隔字线**(DWL)结构简化了译码电路，并且由于把负载分配到多个级上而加快了存取速度。

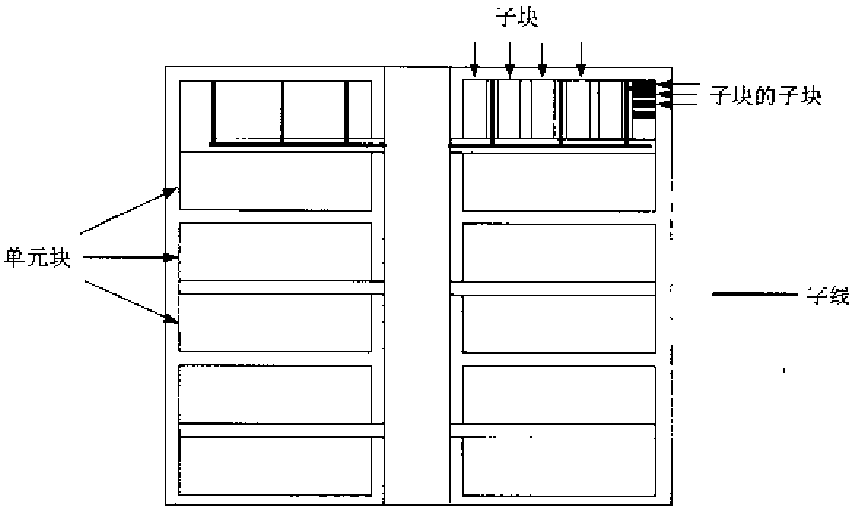


图 13.30 分隔字线结构 RAM 版图的原理图

图 13.31 的逻辑图显示了布线技术。主译码器的输出定义了**全局字线**。这些与块选择信号一起用来激励不同的**块字线**。在层次结构中进入下一个低层次时选择**子块字线**，最终则选择**子-子块**信号。可以设计逻辑门及相应的互连线参数以加速行选择过程。一个简单的方法是应用逻辑努力(Logical Effort)技术使通过每级(逻辑门及输出负载)的延时相同。这可提高速度而不必采用尺寸过分大的晶体管。这一技术可以应用到层次结构中的每个层次。

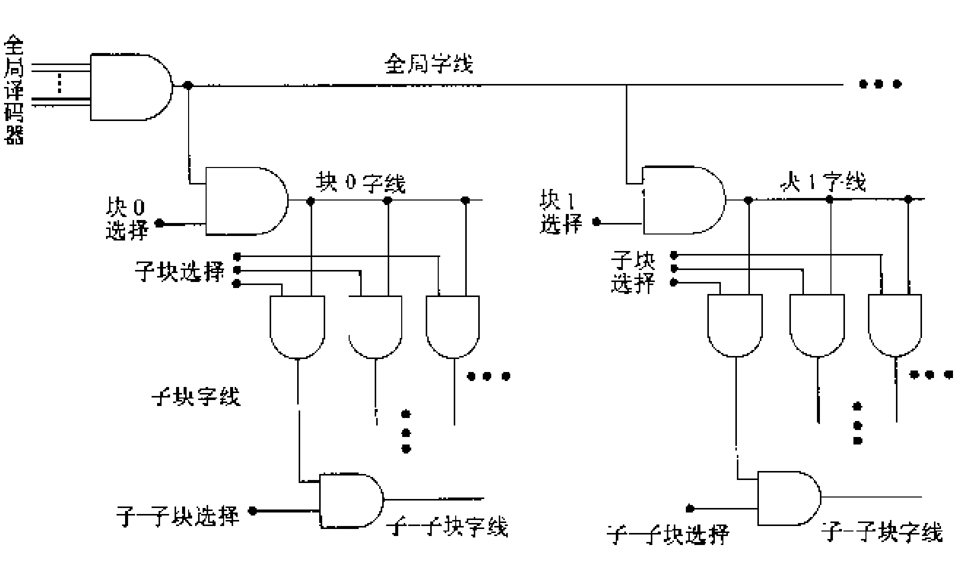


图 13.31 DWL 设计的逻辑图

图 13.32 是采用多块结构的部分 SRAM 的微缩照片。单元块阵列可以由它们规则的版图很容易辨认并且可以清楚看到它们处于下半部。照片的上半部包含译码器、驱动器以及附加电路，其中包括灵敏放大器。

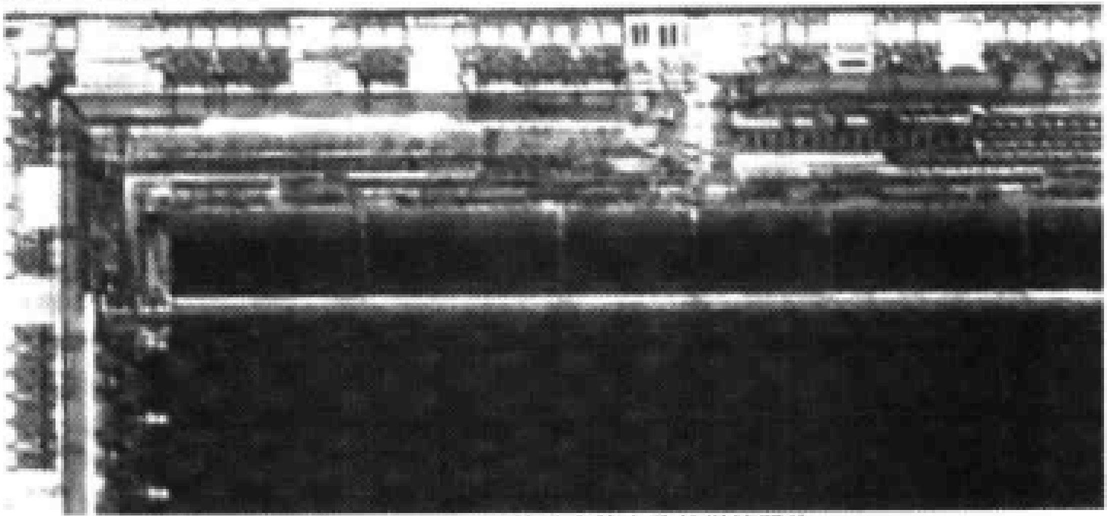


图 13.32 SRAM 块及支持电路的微缩照片

13.4 ROM 阵列

只读存储器(ROM)用于永久的存储。ROM 阵列的结构与 RAM 的结构类似，但单个单元要简单得多。存放在基本只读存储器中的数据是通过有选择地布置 FET 来形成的。由于这是在物理设计时完成的，所以一旦芯片制造出来，数据是不能改变的。

图 13.33 为一个 ROM 阵列，它采用 NOR 门存放 8 位的数据字 $D=d_7d_6d_5d_4d_3d_2d_1d_0$。当把一个地址字送入一个高电平有效的行译码器时，后者把一条字线驱动至高电平而使其他字线仍保持在逻辑 0 电平。字线连到 NOR 门阵列，每行定义一个不同的数据字。多个输入的 NOR 门提供每位 $d_i(i=0,\cdots,7)$ 的数据输出，即每个门的输出由下式确定：

$$d_i = 0 \quad \text{如果任何一个输入为1} \tag{13.23}$$

例如，第 0 行在 7，4，2 和 0 位处连到 NOR 门，对于这些位置输出即为逻辑 1。其余位的位置(6，5，3 和 1)没有连到第 0 行，所以当第 0 行被驱动至高电平时，输出即为逻辑 0(因为其他所有的行都为 0)。因此第 0 行相应于如下的数据字

$$D_{\text{Row 0}} = 01101010 \tag{13.24}$$

每一行都以相同的方式编程。

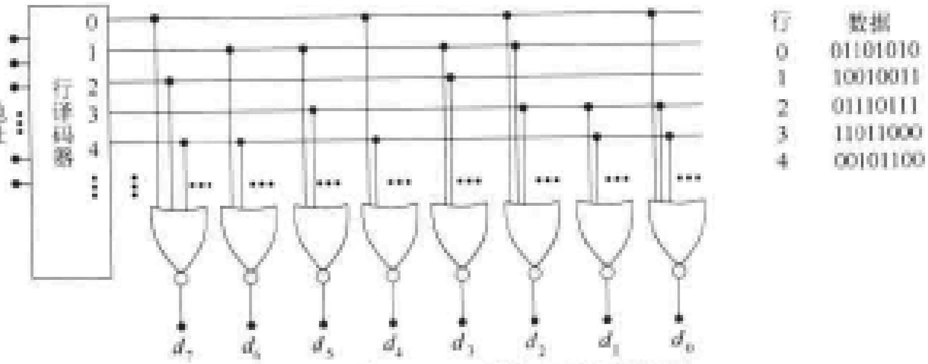


图 13.33 基于 NOR 门的 ROM 逻辑图

图 13.34 为准 nMOS 实现的 ROM。由于每一个 NOR 门只需要一个上拉 pFET,所以编程任务就集中在放置作为下拉器件的 nFET。当 FET 的栅极连至驱动的字线时输出为逻辑 0。这一点很容易理解,即只要注意此时下拉管导通,它提供至地线的连接从而把输出拉低。准 nMOS 电路是有比电路,因此输出低电压值 V_{OL}是由 nFET/ pFET 的比(β_n/β_p)>1 决定的,如在第 9 章中讨论的那样。选择 nFET 的宽长比是一件复杂的事,因为每个下拉管都有一个输入电容 $C_G = C_{ox}WL$,所以如果 W_n 选择得较大以得到较低的 V_{OL},那么字线电容就会增加从而减慢行译码电路。准 nFET 电路的另一个重要特点是每当输出为低即 $V_{out} = V_{OL}$时,它们会有直流功耗。在 ROM 阵列中,只有被选字的输出位为逻辑 0 时才有功耗,这是由于行译码器电路的选择特性使所有其他的 FET 都截止的缘故。

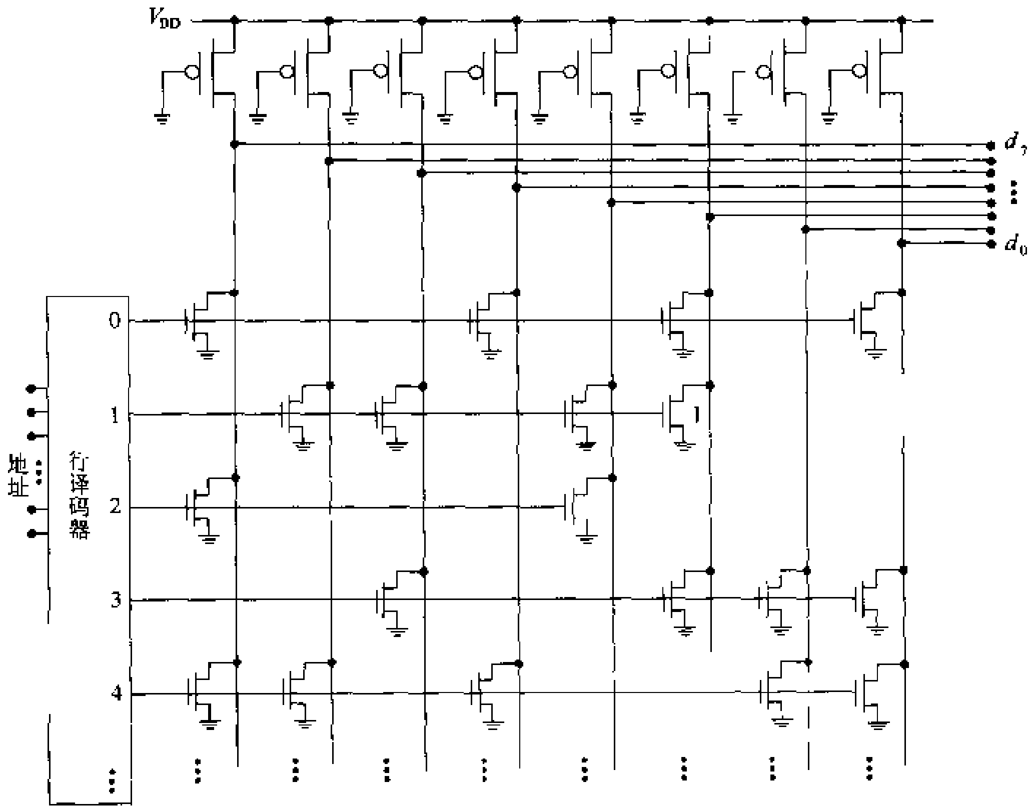


图 13.34 用准 nMOS 电路构成 ROM 阵列

ROM 阵列是版图十分规则的一个好例子。图 13.35 为一个简单的 FET 布置图。它显示了 FET 相对于输入和输出线放置的位置。Metal 1 用于 NOR 门的连接,它为垂直走向(除了 V_{DD}线外),而输出则来自水平方向的 Metal 2 线。这就构成了图 13.36 所示的布局图。通过增加或移去下拉管子可以重新编程。

这一设计可以采用不同的方法来提供基于单元库的 ROM。一种技术是把下拉 nFET 放在字线和 NOR 输出的每个交叉点上;使这条字线通过多晶接触孔连接到 FET 的栅上就相当于把“0”编程到了那个位置上。这就是**掩模编程** ROM 的例子,即存放的数据是由多晶接触掩模来定义的。另一种方法是可以从没有任何 nFET 的阵列开始,运用 CAD 工具按所要求的位置放置管子。

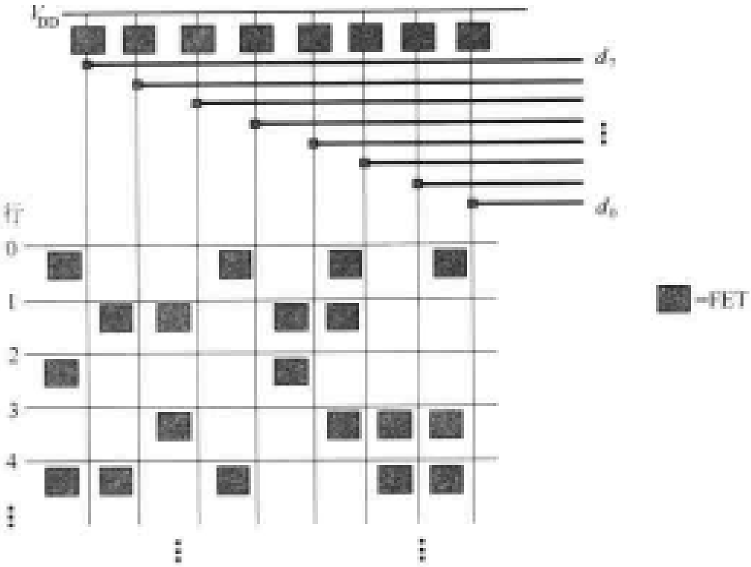


图 13.35　ROM 版图布置

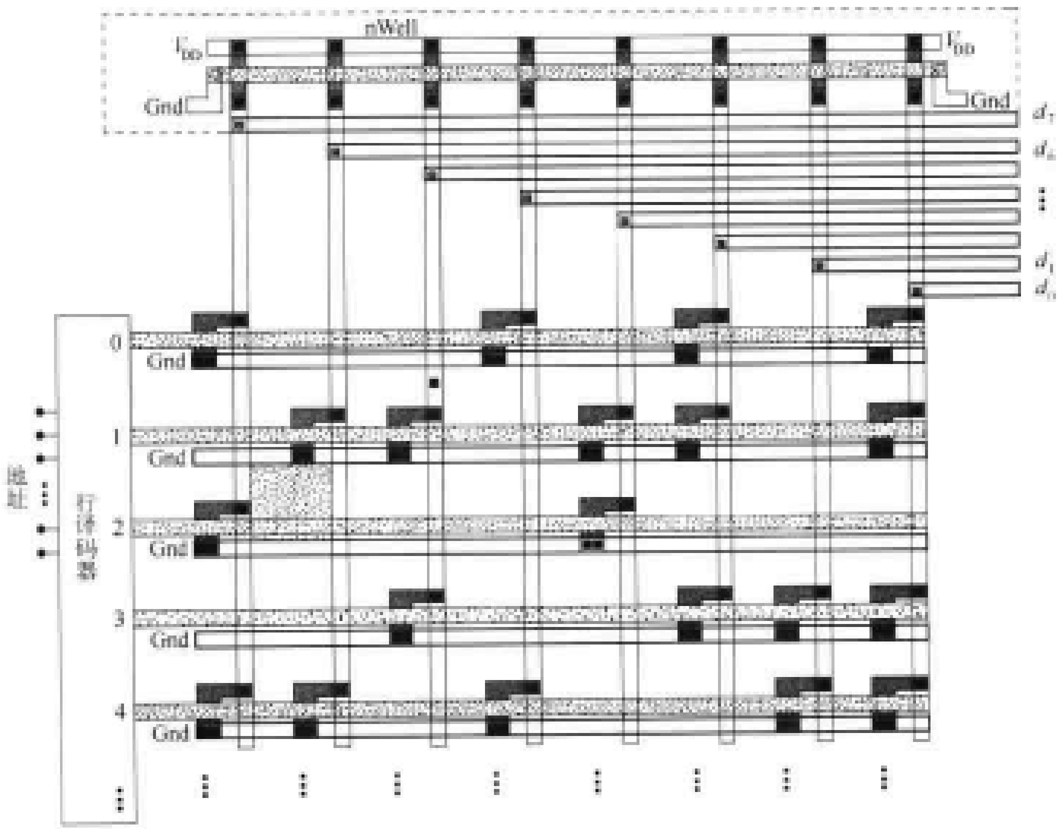


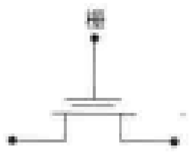
图 13.36　基于 FET 布置的 ROM 布局

用户编程 ROM

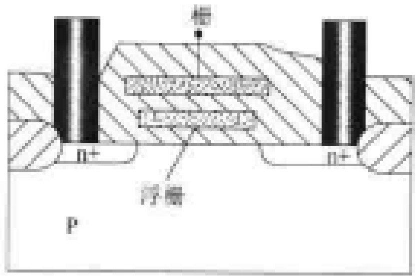
电编程 ROM（PROM）使用户可以根据应用需要存储数据。特殊的电压设置用来进行写单元操作，而读操作是用正常的电压完成的，所以数据可以保持不变。许多 ROM 器件都有擦

除(E)和重写阵列内容的功能。紫外(UV)光擦除用在早期的 EPROM 设计中，但这些器件已为电擦除器件所替代。电擦除 EPROM(即 E^2PROM)用在个人计算机中存放 BIOS，它允许用户更新主板的特性以用在新的设备上。[③]

可重新编程的 ROM 阵列是用特殊的 FET 构成的，它有一对堆叠多晶栅，其电路符号如图 13.37(a)所示。最上面的栅即为通常晶体管的栅极。而另一个多晶栅层则夹在最上面的多晶栅和硅衬底之间，它在电气上与管子或附加电路的任何部分不连接，因而称为浮栅。图 13.37(b)显示了它的细节。浮栅用来存放带负电的电子，它使该管子的阈值电压提高到它的正常值以上。



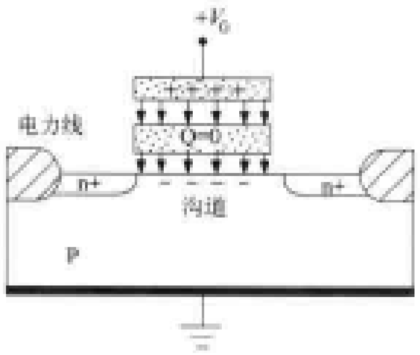
(a) 符号



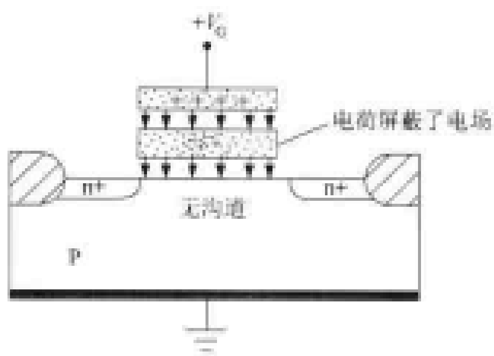
(b) 结构

图 13.37 浮栅 MOSFET

为了理解电荷存储原理及其影响，首先考虑在管子的浮栅上具有 0 电荷。当加上栅电压时就会产生如图 13.38(a)所示的电力线。由于栅是浮空的，所以这一结构的作用类似于一对串联的电容和一个电力线终止在 p 型衬底上的电场。这会产生电子沟道层且使漏源电流流动。在浮栅上 $Q=0$ 使晶体管具有正常的(较低的)阈值电压 V_{Tn}。如果带负电的电子存放在浮栅上，那么电力线将会发生变化如图 13.38(b)所示。当 V_G 为正常值时，在浮栅上的负电荷屏蔽了电力线并阻止它们到达硅表面，于是不会形成沟道，该器件保持在截止状态。若把栅电压提高到一个较高的值 $V_{Tn,H}$，那么 FET 就会起作用。可把该晶体管设计成 $V_{Tn,H}>V_{DD}$，这就保证了当它放在电路中时它总是截止的。



(a) 正常的 V_{Tn} 状态



(b) 提高的 $V_{Tn,H}$ 状态

图 13.38 浮栅上的电荷存储效应

③ BIOS 代表 Basis Input / Output System；BIOS 是在 PC 上电时控制启动过程且把操作系统装入系统存储器。

双阈值电压特点构成如图 13.39 所示的一个 8 位字的 EPROM 电路；为简单起见只画出了读电路。图中具有低(正常)阈值电压 V_{Tn}的 nFET 标有"L"而具有较高阈值电压 $V_{Tn,H} > V_{DD}$的器件标有"H"。一行的栅连接到来自行译码器的字线信号 WL。当 $WL = 1$ 读取这个字时，电压 $V_{WL} = V_{DD}$加到这些逻辑管上。低(L)阈值电压 nFET 导通，于是把输出拉至地(0 V)。另一方面，高(H)阈值电压的管子维持在截止并且通过在位线上的上拉器件产生逻辑 1 的输出电压。对于所显示的例子，输出字为

$$d_7d_6d_5d_4d_3d_2d_1d_0 = 10010110 \tag{13.25}$$

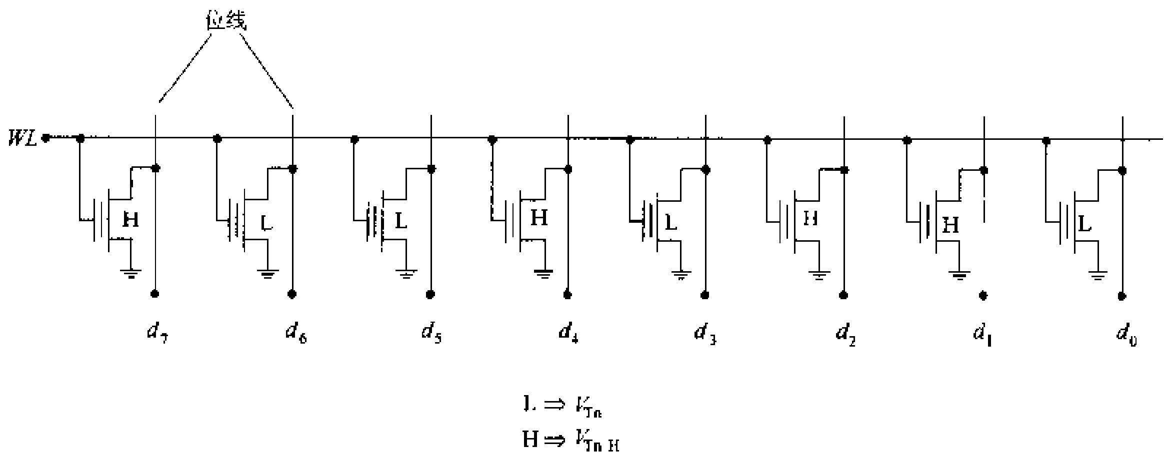


图 13.39　运用浮栅 nFET 的 E^2PROM 字

为了存储随意的数据字，阵列的每个 NOR 门中浮栅 FET 必须连接成下拉器件。

既然看到浮栅 FET 是如何用在电路中的，让我们来考察一下编程技术。在我们一直研究的结构中，电子电荷转移到栅上是通过热电子的量子力学隧穿效应。所谓热电子是具有高能量的沟道电子。为了满足引起隧穿所需要的条件，运用栅电压 $V_{G,prog}$来建立电子沟道，且把一个编程电压加到漏端，如图 13.40 所示。$V_{D,prog}$的典型值在 12～30 V 左右，它引起了一个从漏到源的很强的电场。这个很强的电场加速了到漏端的电子，在那里它们可能发生散射并隧穿至浮栅上。在这些结构中，隧穿电子的保留时间通常估计在 10～20 年的数量级上。

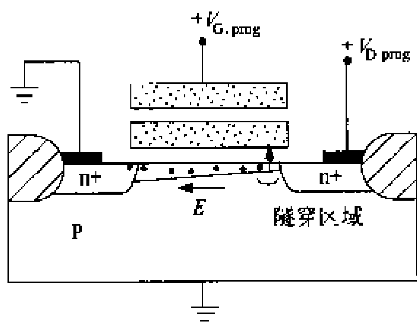


图 13.40　编程一个浮栅 FET

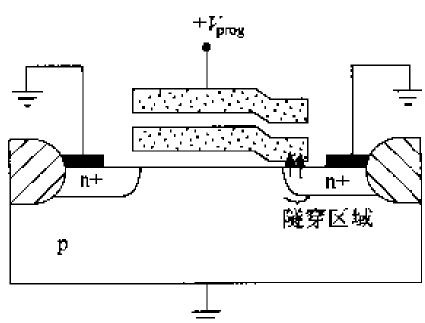


图 13.41　Fowler－Nordheim 隧穿

其他浮栅晶体管的设计采用 **Fowler－Nordheim** 发射原理。它修改了栅的几何形状，使得

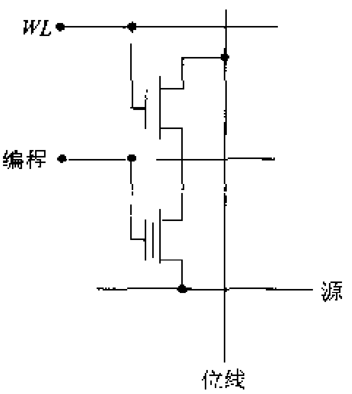


图 13.42　EEPROM 及写入线

浮栅的一部分延伸到 n^+ 漏区，如图 13.41 所示。在栅上加上一个很高的电压使在衬底和栅之间形成一个很强的电场以强化通过氧化层的隧穿。在编程操作期间漏和源都接地。图 13.42 为单元电路完成写操作的情形，此时在编程线上加脉冲，字线 $WL = 1$，而位线和源端都接地。

擦除是通过把所加电压的极性反过来实现的。通常的 EPROM阵列允许位擦除，而 **flash EPROM** 的连接方式是使大块单元同时擦除。后者对存储现代大数据文件，如数字照片产生的数据文件特别有用。

13.5　逻辑阵列

逻辑阵列通常指一个结构化单元，它能被“编程”以提供各种功能和完成系统任务。它们在结构上可以类似于 ROM 但通常应用在更为一般的情形中。在大多数情形中，逻辑阵列的结构是不变的。用户按照已定义的一组规则对阵列进行编程。

13.5.1　可编程逻辑阵列

一个很有用的例子是用来实现 SOP(积项和)逻辑表达式的**可编程逻辑阵列**(PLA)。考虑一组 4 输入变量 a,b,c,d。一个 SOP 函数具有如下形式：

$$f = \sum_i m_i(a, b, c, d) \tag{13.26}$$

式中 $m_i(a,b,c,d)$ 为积项(小项)，它是由输入变量或它们的反信号“与”在一起构成的。在标准形式中，积项包含每一个变量；它的下标 i 的数值是这个字的等效十进制值。就这里的情形，最低层的积项为：

$$\begin{aligned} m_0 &= \bar{a}\cdot\bar{b}\cdot\bar{c}\cdot\bar{d} \quad & m_1 &= \bar{a}\cdot\bar{b}\cdot\bar{c}\cdot d \quad & m_2 &= \bar{a}\cdot\bar{b}\cdot c\cdot\bar{d} \\ m_3 &= \bar{a}\cdot\bar{b}\cdot c\cdot d \quad & m_4 &= \bar{a}\cdot b\cdot\bar{c}\cdot\bar{d} \quad & m_5 &= \bar{a}\cdot b\cdot\bar{c}\cdot d \end{aligned} \tag{13.27}$$

一个 SOP 式是通过把这些积项“OR”(或)在一起得到的，如

$$f = \bar{a}\cdot\bar{b}\cdot c\cdot\bar{d}+\bar{a}\cdot b\cdot\bar{c}\cdot d+a\cdot\bar{b}\cdot c\cdot d+a\cdot\bar{b}\cdot c\cdot d \tag{13.28}$$

这相当于一个 AND-OR(与或)逻辑序列，它形成了 VLSI 实现的基础。AND-OR PLA 的一般结构显示在图 13.43 中。输入被送入到“与”平面以计算出所需要的积项。这些积项然后再送入或(OR)平面中使它们“或”在一起。通过把不同的积项加在一起就可以形成几个 SOP 函数 $f_0, f_1, \cdots$。PLA 的门级方块图显示在图 13.44 中。所选择的 AND 平面通过组合输入提供五个积项。这些积项送入 OR 平面用来形成所示的输出函数。每个输出都可以通过跟踪输入来确定。例如

$$f_x = m_0 + m_4 + m_5 \tag{13.29}$$

可以通过直接考察连接关系而很容易地得到验证。另一个例子是：

$$f_y = m_3 + m_4 + m_5 + m_6 \tag{13.30}$$

应当记住每个积项都是标准式，也就是每个变量以补码或者原码的形式出现在积项中。这些可以直接从功能表中读出。

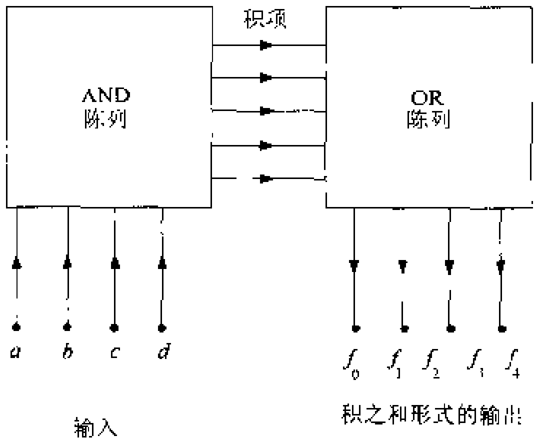


图 13.43　AND－OR　PLA 的结构

最便于使用的一类 PLA 电路是通过每个连接处放置一个 FET，就可以很容易编程的电路。准 nMOS 或动态逻辑类型显然可以解决这个问题，但是它们都存在像对 NAND 门所需要的串联管子确定尺寸的问题。一个解决办法是用基于 NOR 的逻辑链来构成 AND－NOR阵列。图 13.45 (a)显示了所希望的逻辑流。在图 13.45 (b)中对一个 NOR 门采用反相输入可以产生 AND 功能。这可以用 DeMorgan 简化公式从以下的求值中看到：

$$\begin{aligned} g &= \overline{(\overline{a} + \overline{b})} \\ &= a \cdot b \end{aligned} \tag{13.31}$$

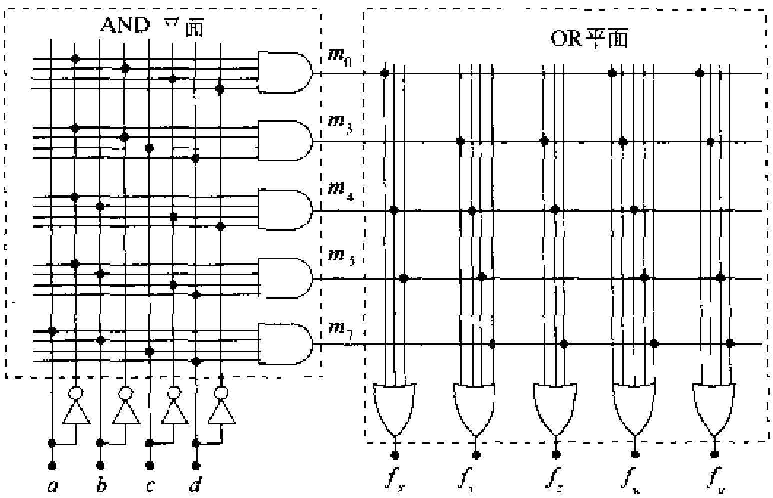
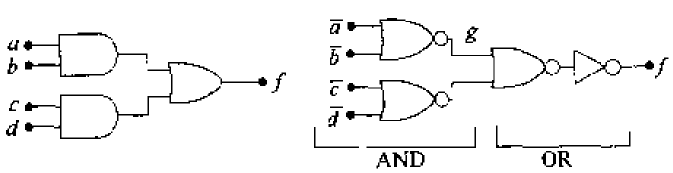


图 13.44　PLA 的逻辑门图



(a) AND－OR 逻辑　　(b) 基于NOR的设计

图 13.45　NOR 门 PLA 逻辑

OR 操作是用 NOR－NOT 链来实现的；或者也可以把反相的 $\overline{f}$ 用来作为输出。基于这一等效关系，一个动态串联链的基本结构显示在图 13.46 的简单例子中。AND 平面由 NOR 门组成，

其输出送入 NOR 门中。反相的输出为：

$$\bar{f}_1 = \overline{a \cdot b \cdot c + \bar{a} \cdot \bar{b} \cdot \bar{c}}$$
$$\bar{f}_2 = \overline{a \cdot b \cdot c} \tag{13.32}$$

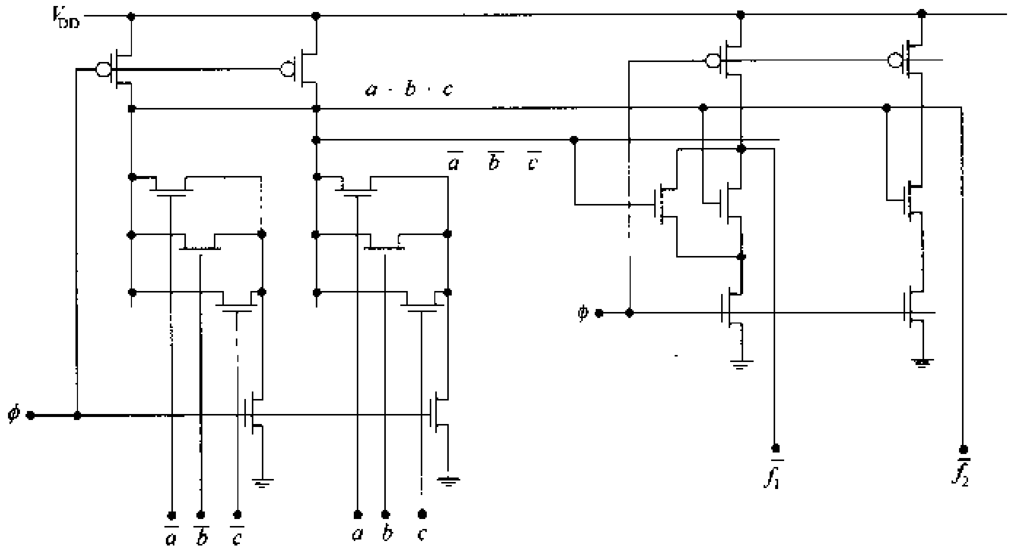


图 13.46 基于 NOR 门的动态 CMOS PLA

其他逻辑函数可以通过在其中一个平面或同时在两个平面中扩展逻辑门来构成。单个时钟 ϕ 用来同步这两个平面的操作。这些可以被分成不同的两部分信号，其中输入平面稍微领先于输出平面。在本设计中，nFET 的尺寸决定了这两个部分门从高到低的时间。

静态准 nMOS 设计可以通过取消求值 nFET 且使 pFET 门接地来实现。由于准 nMOS 电路是有比电路，输出电压 V_{OL} 取决于 (β_n/β_p) 的值，但 NOR 门的设计是很容易的。另一个问题是在 $V_{out} = V_{OL}$ 时发生的直流功耗。PLA 的另一种设计方法是设计一个基于 NOR 的 OR - AND 阵列来实现“和项积”(POS)函数。它运用了简单的逻辑且是习题[13.12]的内容。

这个方法的可编程性特点可以从图 13.47 中看出。任何一个逻辑平面都可以表示成一个 NOR 门阵列。每个门的输出是由 nFET 逻辑块决定的。设计者对该阵列编程是通过把 FET 按要求放置并连接到每个门中。物理设计的特点是把规则的 FET 图案放在预充电 - 求值管之间。

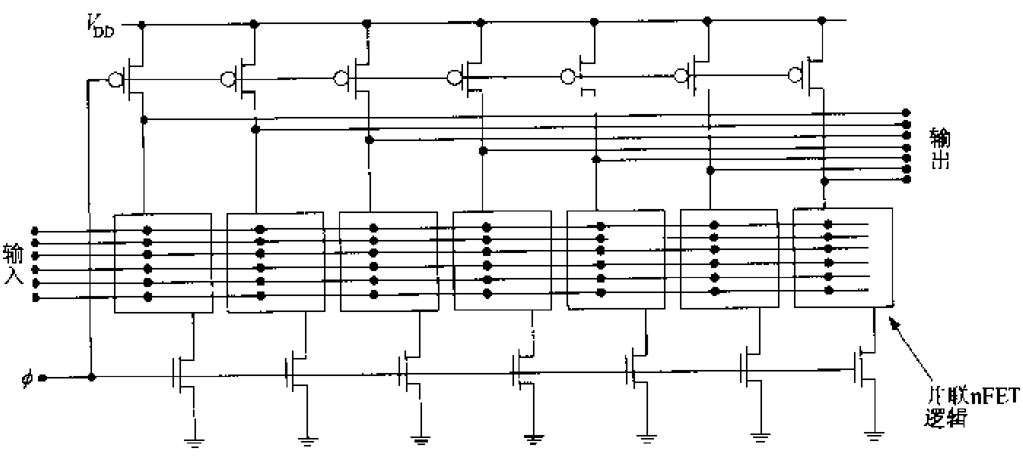


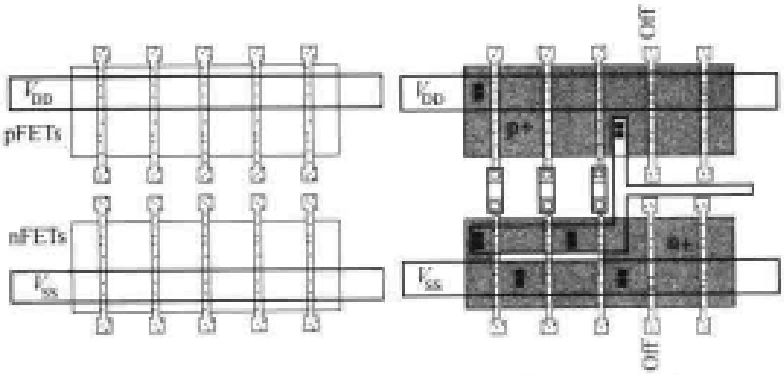
图 13.47 一般化的基于 NOR 的逻辑平面

13.5.2　门阵列

门阵列这个名字用来描述一整类器件。它通常是指用户可以按要求设置逻辑的可编程芯片。编程技术随芯片本身结构的不同而不同。

在物理设计层，一个门阵列就其名义包括一个逻辑门的阵列，这些逻辑门在互连线掩模层上通过导线连接在一起。这类器件的基础是如图 13.48(a)所示的预先定义的晶体管的“海洋”。公共的 n+ 或 p+ 区域分别用来确定 nFET 和 pFET 串。V_{DD}和 V_{SS}金属线的位置是已知的，但在未经编程的阵列中没有与电源的任何连接；这些连接是通过接触孔掩模加上去的。一般的布线是通过同时在金属层掩模和接触孔掩模上增加特征细节来完成的。

逻辑门可以用任何设计类型构成，但组合静态设计是最容易和最普通的。图 13.48(b)为一个 NOR3 门的连线。电源接触已加入到 V_{DD}和 V_{SS}线上。每对互补的 nFET/pFET 是通过金属线和多晶接触孔用导线连接的。采用金属线和有源接触孔实现的管子连线形成了三个并联的 nFET 和三个串联的 pFET。在这类阵列中的电气绝缘是通过截止管子实现的。在 NOR3 门中，逻辑 pFET 与其他 p 沟管子间是通过把栅连至电源 V_{DD}来绝缘的；将 nFET 的栅连至 V_{SS}对 nFET 串也产生了相同的绝缘效应。较先进的工艺采用氧化物绝缘以防止大的泄漏电流。这可以得到比较稳定的电气特性，但会使布线更为复杂，除非存在公共的 n+/p+ 串。



(a) 门阵列基片　　　　(b) NOR3 门的布线

图 13.48　门阵列中的管子排布

门阵列允许采用半定制逻辑迅速地实现原型。由于设计者可以涉及各个管子，所以对于要实现的电路或逻辑没有任何预先的限制。这一过程可以高度自动化，因为门的变换关系定义得很明确。一个缺点是最终的连线由掩模阶段确定，所以圆片必须经过一部分生产线加工。这增加了设计的周期。另一个问题可能来自电气特性。在一个均匀的“门”海中，每种极性 FET 的宽长比是相同的。由于不可能调整尺寸，所以忽视了对管子尺寸敏感的某些电路。大的(W/L)值是很普遍的，它允许用简单的方式设计大驱动能力的电路。因为连线寄生参数，特别是线电容随长度增加，所以开关速度也许是一个问题。

门阵列设计会使所实现的芯片比用定制设计尺寸时要大。但通常并不考虑这一点，因为主要的目的是通过在 FET 层次上半定制地控制电路以实现较快的设计过程。

现场可编程门阵列(FPGA)非常普遍地用来测试逻辑设计和制造有限数量的产品。正如

其名，FPGA 可以"在现场"编程，而不必经过实验室或生产线。PC 的插接板允许用户定义逻辑并以快速简单的步骤去"烧"电路。测试一个系统的逻辑设计变成一个比较简单的任务。现代的 FPGA 密度很高，可以包含几百万门。甚至像 32 位流水线微处理器这样的复杂系统也可以运用 FPGA 进行硬件仿真。

FPGA 设计成通用的逻辑电路，用户可以定义要用哪些元件以及它们是如何连接在一起的。编程通常是一个电气过程，它在物理层次上采用内建在此结构中的反熔丝来实现。图 13.49 (a)显示了一个反熔丝。根据定义，一个反熔丝器件通常像是在两个导电层 1 和 2 之间的一个开路。反熔丝通过使大电流通过它而"熔断"，于是反熔丝层被熔化而在 Metal 1 和 n+ 之间形成低阻的接触。这可以应用在金属层与 n+/p+ 有源区之间，如图 13.49 (b)所示；或在金属层和多晶层之间，如图 13.49 (c)所示。

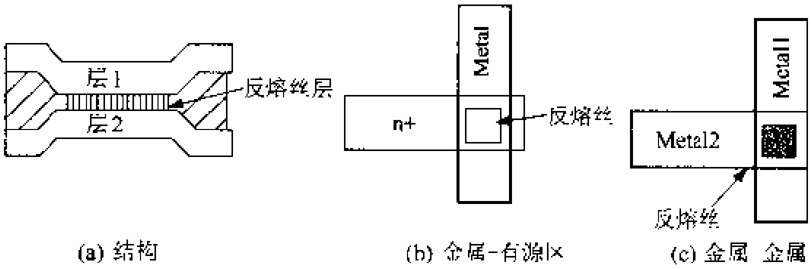


图 13.49 可编程反熔丝结构

FPGA 中的基本电路随供应商而有所不同。最一般的方法是所提供的逻辑块包含像组合逻辑，锁存器和/或触发器，多路选择器以及查表这样的元件。通过 CAD 工具及覆盖许多标准情形如有限状态机或并行处理器的设计说明使系统设计变得十分方便。FPGA 设计的细节越出了本书的范围，但有几本优秀的参考书介绍这方面的内容。Smith 的书(参考资料[10])介绍许多不同的商业产品，其特点几乎像是一本百科全书。参考资料[3]对这个内容做了可读性很好的介绍。从网站来的和印刷出来的供应商数据是非常丰富的，它们通常还包括许多辅导资料以帮助潜在的用户。

13.6 参考资料

[1] R. Jacob Baker, Harry W. Li, and David E. Boyce, **CMOS Circuit Design, Layout and Simulation**, IEEE Press, Piscataway, NJ, 1998.

[2] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[3] Stephen D. Brown, Robert J. Francis, Jonathan Rose, and Zvonko G. Vranesic, **Field-Programmable Gate Arrays**, Kluwer Academic Publishers, Norwell, MA, 1992.

[4] Michael D. Ciletti, **Modeling, Synthesis and Rapid Prototyping with the Verilog HDL**, Prentice Hall, Upper Saddle River, NJ, 1999.

[5] Paul R. Gray and Robert G. Meyer, **Analysis and Design of Analog**

Integrated Circuits, 3rd ed., John Wiley & Sons, New York, 1993.

[6] Ken Martin, **Digital Integrated Circuit Design**, Oxford University Press, New York, 2000.

[7] Betty Prince, **High Performance Memories**, John Wiley & Sons, Ltd., West Sussex, U.K., 1999.

[8] Betty Prince, **Semiconductor Memories**, 2nd ed., John Wilcy & Sons, Ltd., West Sussex, U.K., 1991.

[9] Jan M. Rabaey, **Digital Integrated Circuits**, Prentice-Hall, Upper Saddle River, NY, 1996.

[10] Michael J. S. Smith, **Application-Specific Integrated Circuits**, Addison-Wesley, Reading, MA, 1997.

[11] John P. Uyemura, **A First Course in Digital System Design**, Brooks-Cole, Monterey, CA, 2000.

[12] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[13] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.

13.7 习题

[**13.1**] 采用晶体管级的基本单元建立图 13.2 所示两个 SRAM 单元的 Verilog 开关级模块。

[**13.2**] 写出存放以下数据的存储阵列的 Verilog 描述。

(a) 在 2048 个位置上存放 32 位的字。

(b) 16KB。

(c) 在 8K 位置上存放 8 位的字。

[**13.3**] 假设一个 6T SRAM 单元采用 β 比率为 1 的设计。是否即便 $R_A = R_n$ 仍有可能写这个单元？用电路来解释你的回答。

[**13.4**] 考虑图 13.31 所示分隔字线结构。运用 Logical Effort 的概念，解释为什么这个设计可能比采用单根字线的设计要快？以定量方式运用方程来列出你的理由。可以对电容值等做出必要的假设，并且为简化起见在你的分析中采用 $r = 2$。

[**13.5**] 设计一个逻辑电路，使在图 13.7 的双口 SRAM 单元中避免写冲突。假设两个端口有相同的优先权，重点放在避免同时发生写操作。

[**13.6**] 在 DRAM 中存储电容的值 $C_s = 55$ fF。该电路限制电容电压值为 $V_{max} = 3.5$ V。当存取管截止时，这个单元的漏电流估计为 75 nA。

(a) 在 C_s 上可以存放多少个电子？

(b) 由于漏电流在 1 秒内有多少个基本电荷单位 q 离开这个单元？

(c) 计算使存储电荷数减少到 100 个所需要的时间。

[**13.7**] 一个 DRAM 单元的存储电容 $C_s = 45$ fF。它用在 $V_{DD} = 3.3$ V 和 $V_{Tn} = 0.55$ V 的系统中。位线电容 $C_{bit} = 250$ fF。

(a) 求在 C_s 上能存储的最多电荷数量。

(b) 假设电容的电压被充电到 V_{max} 的电平。在时间 $t = 0$ 时控制存取 FET 的字线降至 $WL = 0$ 值。漏电电流估计为 50 nA，为了检测逻辑 1 状态，在位线上的电压必须至少为 1.5 V。求保持时间。

[13.8] 考虑一个 DRAM 单元,它的存储电容 $C_s = 55$ fF。电源电压 $V_{DD} = 3.0$ V,而存取 FET 的阈值电压 $V_{Tn} = 0.65$ V。存储电容的漏电流估计为 250 pA,而位线电容 $C_{bit} = 420$ fF。在时间 $t = 0$ 字线被拉为低电平时,电容两端的电压为 V_{max}。在时间 $t = 10$ ms 时通过把字线提高到 $WL = 1$ 而开始读操作。求在位线上的电压。

[13.9] 设计一个具有如下输出的 AND－OR PLA:

$$\begin{aligned} f_1 &= m_0 + m_2 + m_6 \\ f_2 &= m_0 + m_1 + m_4 + m_5 + m_6 \\ f_3 &= m_3 + m_4 + m_7 \end{aligned} \tag{13.33}$$

画出逻辑方块图,然后把它转变成与 CMOS 兼容的电路。

[13.10] 设计一个提供如下输出的 OR－AND PLA:

$$\begin{aligned} F_1 &= M_2 \cdot M_3 \cdot M_5 \\ F_2 &= M_0 \cdot M_1 \cdot M_4 \\ F_3 &= M_1 \cdot M_2 \cdot M_6 \cdot M_7 \end{aligned} \tag{13.34}$$

可以采用什么常用的 CMOS 门来实现这一阵列的物理设计?

[13.11] 基于 NOR 门设计一个动态 CMOS AND－OR PLA。设计这个电路使输入为 a, b, c,而输出为:

$$\begin{aligned} g_1 &= (a \cdot \bar{b} \cdot c) + (\bar{a} \cdot \bar{b} \cdot \bar{c}) \\ g_2 &= (\bar{a} \cdot \bar{b} \cdot c) + (a \cdot b \cdot c) \\ g_3 &= (a \cdot \bar{b} \cdot \bar{c}) + (a \cdot \bar{b} \cdot c) \\ g_4 &= (\bar{a} \cdot \bar{b} \cdot \bar{c}) + (a \cdot \bar{b} \cdot \bar{c}) \end{aligned} \tag{13.35}$$

[13.12] 设计一个动态 CMOS NOR－NOR PLA,其输入为 a, b, c,输出如下的 POS 函数:

$$\begin{aligned} f_1 &= (a + \bar{b} + c) \cdot (\bar{a} + b + c) \\ f_2 &= (\bar{a} + b + \bar{c}) \cdot (a + b + c) \\ f_3 &= (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c}) \end{aligned} \tag{13.36}$$

首先构成逻辑电路,然后设计电子电路。

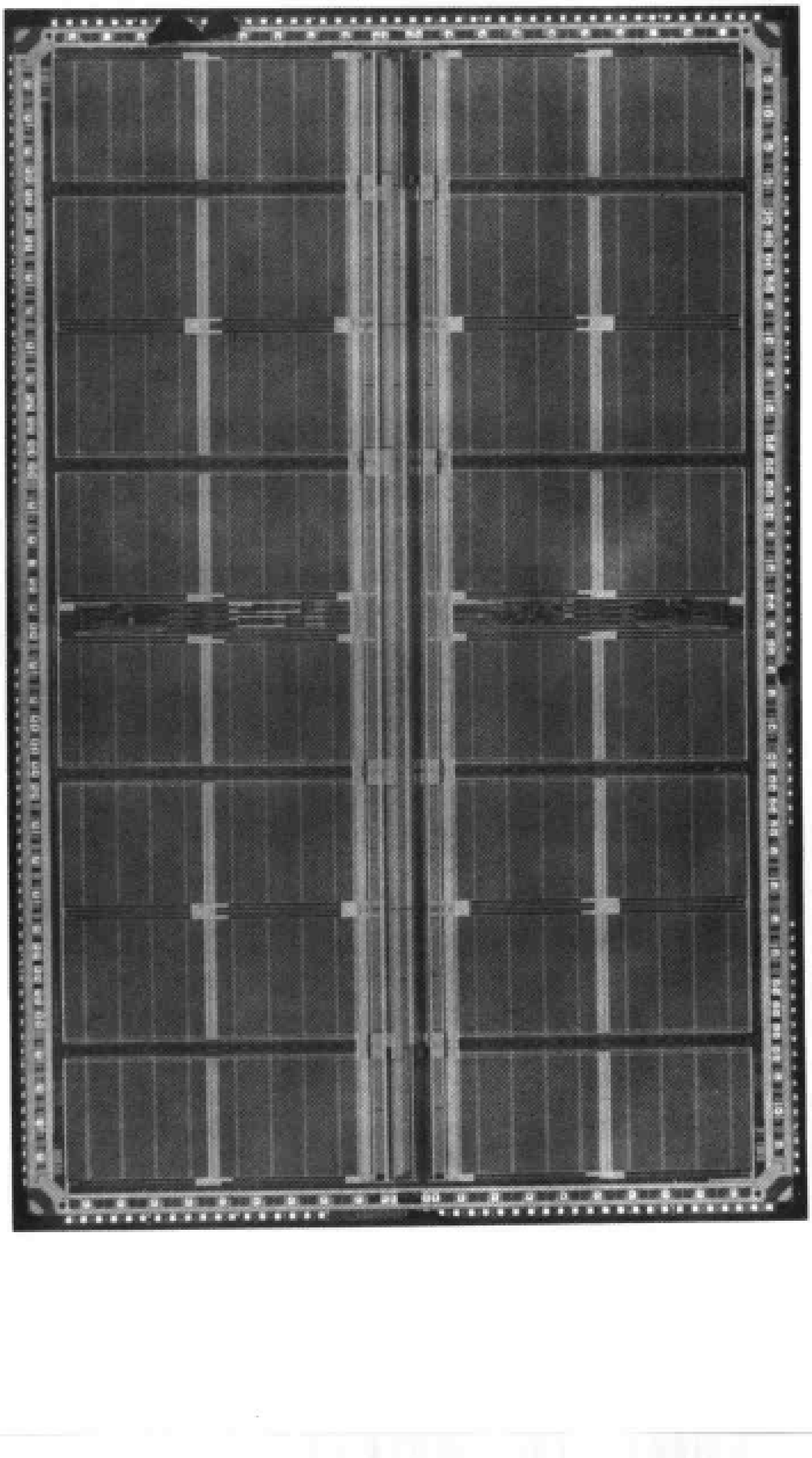
[13.13] 设计一个含有如下数据的 FET 可编程 ROM。

地址	数据
0	0100
1	1111
2	1010
3	0001
4	1011
5	0111
6	1110
7	1001

(13.37)

[**13.14**]　考虑图 13.48(a)所示的一般的门阵列基片。设计连线以构成采用标准串并联结构的具有如下功能的复合逻辑门：

$$F=\overline{a\cdot b\cdot c+a\cdot e}\tag{13.38}$$



第 14 章　系统级物理设计

CMOS VLSI 设计涉及硅片电路的物理特性。一个高层次的体系结构功能可以用若干不同电路中的任何一个来实现,但各功能块必须用导线连在一起以完成芯片设计。在本章中将考察宏观物理设计的特点。

14.1　大规模集成电路的物理设计

至今我们集中在研究采用 CMOS 工艺的较为简单的逻辑功能。一旦建立了单元库,它可以用来把逻辑单元例举到主设计中以构成复杂的 VLSI 系统。这个自下向上的描述代表从低层次的基本单元转向高层次的系统。与这个设计层次上的改变相联系的是,当重点从位一级的微尺寸结构上升到较大尺寸的单元和部件时所引起的一些物理考虑。互连线、信号分布以及大的电路模块的特性只是必须考虑的几个问题。因为芯片的性能最终取决于它的部件,因此物理设计和版图在大规模情况下的特点是极为重要的。若把一个快速的开关电路嵌入很慢的互连线网中,则抵消了在低层次设计中获得的速度。

说明某些重要问题的一个典型的自上而下设计流程显示在图 14.1 中。在系统的 HDL 描述写出并且验证之后,逻辑综合提供了第一步设计。然后需要对所设计的电路进行模拟,重点是确保这个设计的行为如同所希望的那样。设计顺序中的下一步则是布局、单元布置、时钟路径以及信号布线等的物理设计。这些即是考虑解决设计芯片的大规模问题,也是本章的内容。一个高速数字系统的功能是与整个时序紧密联系在一起的:如时钟的分布,门的延迟,锁存以及其他考虑。时序信息提供了一个反馈路径,它把每个连续的设计层次与下个层次联系起来。测试问题也在设计阶段考虑,如图 14.1 中左边所示。由于测试 VLSI 芯片的问题对工业界是非常重要的,所以本书后面专门有整章来介绍它。

硅片集成电路的尺寸随电路的复杂性和晶体管的数目而增加。整个来说,我们的目标是设计一个芯片具有最小的面积 A_{chip} 以同时提高每个圆片的芯片数 N 和整个成品率 Y。对于给定的工艺和圆片尺寸,目的也许是为这个设计选择最小标准尺寸的芯片。但不论实际的芯片尺寸有多大,当设计整个芯片时会产生许多与物理设计相关的问题。本章将介绍最为重要的问题。其中某些问题,如互连线延时和布线,则被认为是未来设计的主要障碍。

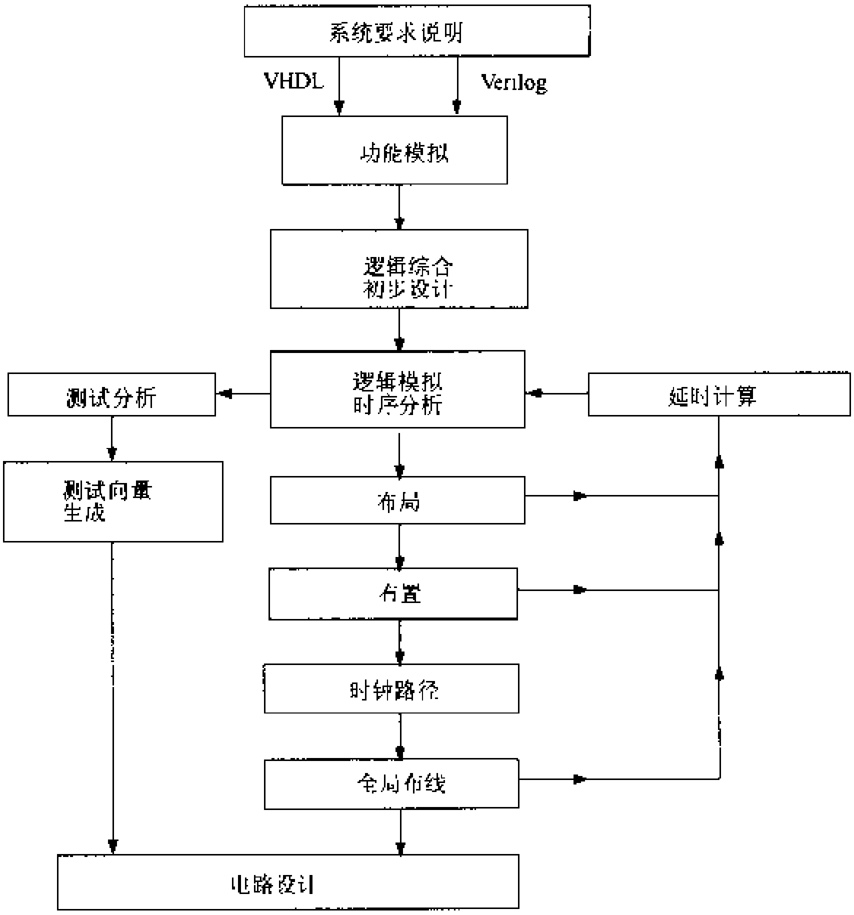


图 14.1　显示芯片级物理设计问题的设计流程

14.2　互连线延时模型

在高密度 VLSI 中最关键的问题之一是考虑互连线的问题。它们会引起影响系统时序的信号延时，并且常常会引起极为复杂的版图布线问题。我们已经看到许多在生产过程中是如何对互连线进行特别处理的：硅化物多晶线，多层金属互连，以及使用铜。在本节中，将建立起等效电路为互连线延时提供数学模型。布线技术将在 14.5 节讨论。

分析的起点是简单的绝缘互连线，如图 14.2 所示，它代表了芯片上任何材料层。这条线的尺寸为长度 l，宽度 w 及厚度 t。从 In 至 Out 的线电阻 R_{line} 由下式给出：

$$R_{line} = R_s\left(\frac{l}{w}\right)\ \Omega \tag{14.1}$$

式中，R_s 为该材料层的薄层电阻，(l/w) 为所包含的尺寸为 $(w \times w)$ 的方块的数目。定义每单位长度的电阻 r 如下：

$$r = \frac{R_s}{w}\quad \Omega/\text{cm} \tag{14.2}$$

方程(4.1)变为：

$$R_{line} = rl \tag{14.3}$$

这表明寄生电阻随线长 l 增加的方式很容易处理。显然，高阻材料层如多晶硅比起低阻金属层来会有较多的问题。

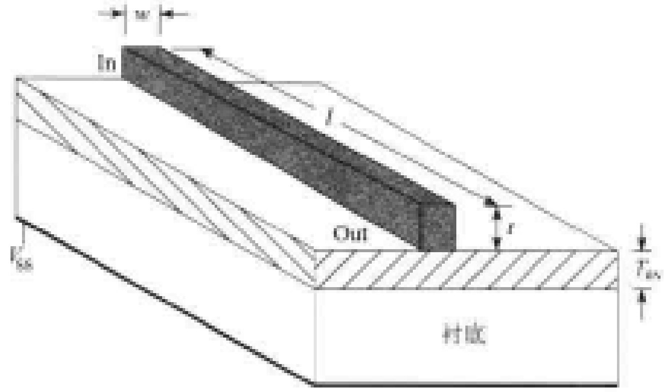


图 14.2 绝缘互连线

总的线电容可用简单的平行板公式来估算：

$$C_{line} = \frac{\varepsilon_{ox} l w}{T_{ox}} \text{ F} \tag{14.4}$$

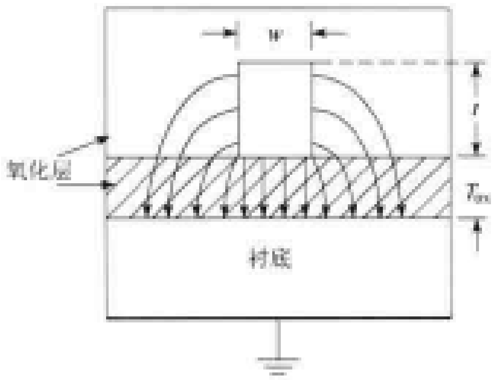


图 14.3 一根绝缘互连线的电力线

式中，T_{ox}是在连线与衬底之间绝缘氧化层的厚度。C_{line}也称为线的**自身电容**。尽管这个方程提供了初步的估计，但它却忽略了当连线处在正电压时边沿和侧面的寄生电场的影响；图 14.3 显示了对边缘场的最初修正。考虑这些影响的每单位长度电容 c 的一个经验公式为[10]

$$c = \varepsilon_{ox}\left[1.15\left(\frac{w}{T_{ox}}\right) + 2.8\left(\frac{t}{T_{ox}}\right)^{0.222}\right] \text{ F/cm} \tag{14.5}$$

于是

$$C_{line} = cl \tag{14.6}$$

为总的电容值(法拉)。从概念上说，第一项考虑了连线底部的边缘效应，第二项取决于厚度 t 并且来自于侧面效应。在 CMOS 芯片中，最大的连线电容值发生在离衬底最近的工艺层上。

【例 14.1】 考虑第一层金属互连线，它的横截面尺寸为 $w=0.35\ \mu\text{m}$ 及 $t=0.7\ \mu\text{m}$，布置在厚度为 $T_{ox}=0.9\ \mu\text{m}$ 的氧化层上。每单位长度的电容为：

$$\begin{aligned} c &= (3.9)(8.854\times10^{-14})\left[1.15\left(\frac{0.35}{0.9}\right) + 2.8\left(\frac{0.7}{0.9}\right)^{0.222}\right] \\ &= 1.07 \text{ pF/cm} \end{aligned} \tag{14.7}$$

如果薄层电阻为 $R_s = 0.02\ \Omega$，则每单位长度的电阻为

$$r = \frac{0.02}{0.35\times10^{-4}} = 571 \quad \Omega/\text{cm} \tag{14.8}$$

一根长度为 $l = 40\ \mu\text{m}$ 互连线的特征参数为：

$$\begin{aligned} R_{\text{line}} &= (571)(40\times10^{-4}) = 2.29\ \ \Omega \\ C_{\text{line}} &= (1.07)(40\times10^{-4}) = 4.28\ \text{fF} \end{aligned} \tag{14.9}$$

若长度增加为 $l = 225\ \mu\text{m}$ 则

$$\begin{aligned} R_{\text{line}} &= (571)(225\times10^{-4}) = 12.85\ \ \Omega \\ C_{\text{line}} &= (1.07)(225\times10^{-4}) = 24.1\ \text{fF} \end{aligned} \tag{14.10}$$

虽然电阻保持相对较小(因为 R_s 很小)，寄生线电容则与 MOSFET 的值处于相同的数量级，这使它对于分析非常重要。

计算 R_{line} 和 C_{line} 的值使我们可以建立电路模型以研究参数的影响。模拟连线的最简单方法是建立简单的两元件电路，如图 14.4 所示，以包括从 In 至 Out 连线的影响。由于它的画法，它称为"单级梯形电路"。如果输入电压 $v_i(t)$ 以阶跃方式从 0 变化到 1(反之亦然)，那么输出电压 $v_o(t)$ 的变化延迟了一个时间常数

$$\tau = R_{\text{line}} C_{\text{line}} \tag{14.11}$$

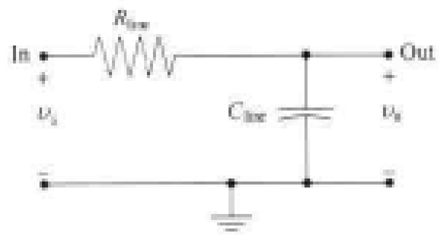


图 14.4 单级梯形模型

它即为对互连线影响信号传输的低阶估计。必须更进一步地深入研究寄生参数的来源，以便理解为什么这是一个粗略的估计。我们的研究也将推导出更为精确的时间常数值。

考虑一根互连线沿 z 方向延伸如图 14.5 所示。在 $z=0$ 处加入输入电压 $v_i(t)$ 而输出端处为 $z=l$。若给输入电压一个脉冲，使 $v_i \to V_{DD}$，那么驱动源必定驱动电流通过电阻线而同时又给它遇到的线电容充电。实际上，充电开始于 $z=0$ 处并且沿 z 增加的方向逐渐向右，所以连线电压本身实际上同时是位置 z 和时间 t 的函数。为此，把它表示成 $v(z,t)$。连线的电阻和电容都具有增量的特性，因此采用集总单元的值 R_{line} 和 C_{line} 来近似它们的值，从根本上会限制分析的精确性。

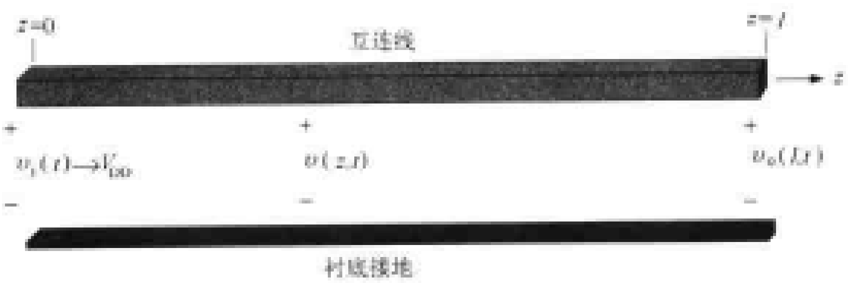


图 14.5 互连线的物理模型

有两种技术可以用来模拟这个问题。第一种方法是把连线分成几个 RC 梯形级，近似寄生参数的分布特性。图 14.6 说明了这一步骤。连线电阻 R_{line} 和连线电容 C_{line} 被分隔成 m 段，每段的值为

$$R_m = \frac{R_{line}}{m} \quad , \qquad C_m = \frac{C_{line}}{m} \tag{14.12}$$

然后可用它们构成多级梯形电路。图 14.6(a)为 $m=1$ 的情形。这一电路由参照的时间常数来定义：

$$\begin{aligned}\tau_1 &= R_1 C_1 \\ &= R_{line} C_{line}\end{aligned} \tag{14.13}$$

图 14.6(b)中两级梯形电路的时间常数可以用 Elmore 公式分析如下：

$$\tau_2 = C_2(2R_2) + C_2(R_2) = 3R_2C_2 \tag{14.14}$$

类似地，图 14.6(c)中的三级梯形电路由以下的时间常数表达式描述：

$$\tau_3 = C_3(3R_3) + C_3(2R_3) + C_3(R_3) = 6R_3C_3 \tag{14.15}$$

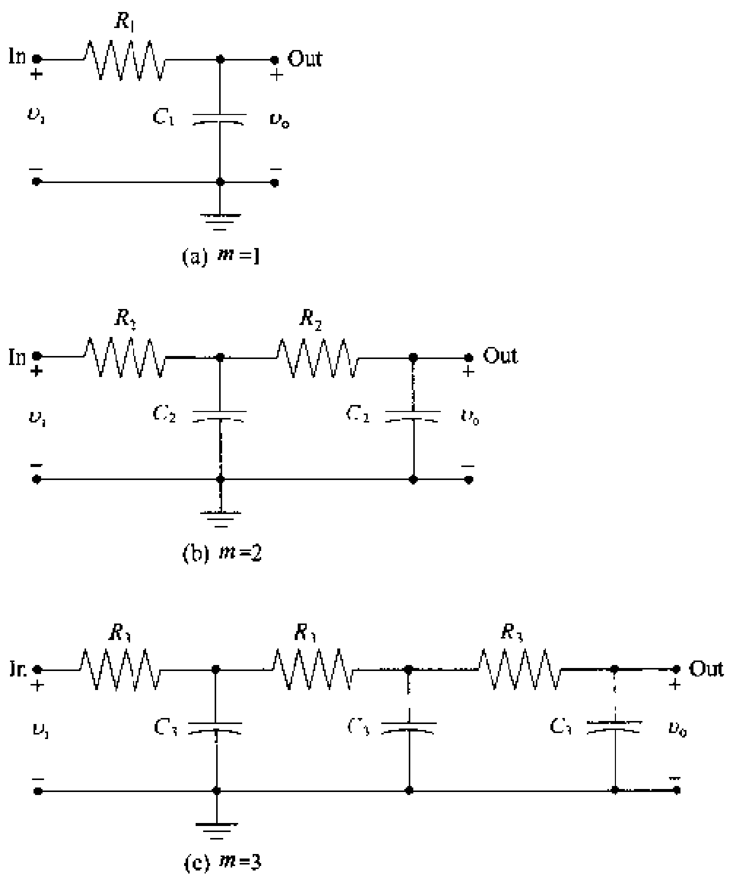


图 14.6　多级梯形电路

一般地，一个 m 级梯形电路的时间常数为

$$\tau_m = \frac{m(m+1)}{2} R_m C_m \tag{14.16}$$

把方程(14.12)的关系代入这个表达式中得到了如下用总的连线电阻和电容表示的时间常数：

$$\tau_m = \frac{m(m+1)}{2}\left(\frac{R_{\text{line}}}{m}\right)\frac{C_{\text{line}}}{m} = \frac{m(m+1)}{2m^2}R_{\text{line}}C_{\text{line}} \tag{14.17}$$

对于大的 m 值，它的极限形式为：

$$\tau = \tau_m \to \frac{1}{2}R_{\text{line}}C_{\text{line}} \tag{14.18}$$

此即为该连线总的时间常数 τ。把它重写为：

$$\tau = R_{\text{line}}\left(\frac{C_{\text{line}}}{2}\right) \tag{14.19}$$

得到图 14.7 集总元件的"π"网络。① 它把($C_{\text{line}}/2$)放置在连线的每一端来均衡连线电容的影响。这一简化的互连线模型常用在实际中，以得到连线延时合理的一阶估计。

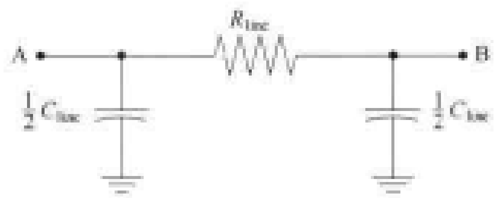


图 14.7 简单的 RC 互连线模型

图 14.8 为应用这个模型计算信号延时的例子。图 14.8(a)中的互连线连接两个逻辑门。驱动门产生的输出必须传送到负载门。采用"π 模型"建立如图 14.8(b)的等效电路，图中定义了电压 $v(t)$ 和 $v_L(t)$，分别相应于驱动门的输出和负载值。负载电容 C_L 代表总的负载电容，它由下式给出：

$$C_L = \left(\frac{1}{2}\right)C_{\text{line}} + C_{\text{in}} \tag{14.20}$$

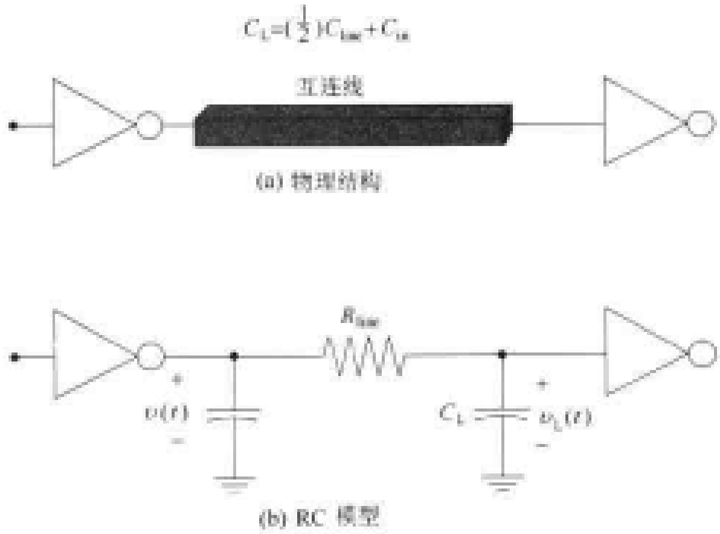


图 14.8 用互连线模型估计信号延迟

其中，C_{in}是负载门的输入电容。注意 $v(t)$定义为左边电容两端的驱动电压。由连线引起的信号延时的计算可以通过说明驱动电压 $v(t)$然后计算出响应 $v_L(t)$。

① 之所以称为"π"网络是因为元件的安排形似这个希腊字母。

假设对 $v(t)$ 采用脉冲输入如图 14.9 所示。在时间 $t=0$ 时，这一电压以阶跃方式从 0 变化到 V_{DD}。这一过渡过程可以写成：

$$v(t) = V_{DD}u(t) \tag{14.21}$$

式中 $u(t)$ 是单位阶跃函数，定义为：

$$u(t) = \begin{cases} 0 & (t<0) \\ 1 & (t\geqslant 0) \end{cases} \tag{14.22}$$

这一电路的阶跃响应为

$$v_L(t) = V_{DD}[1-e^{-(t/\tau)}] \tag{14.23}$$

式中时间常数为

$$\tau = R_{line}C_L \tag{14.24}$$

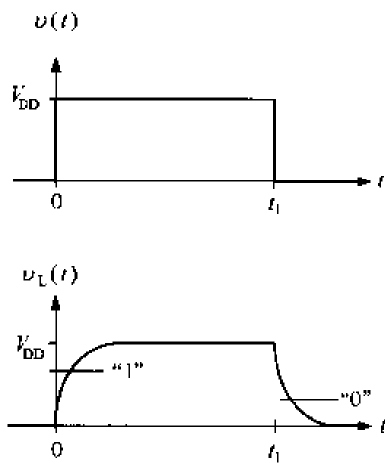


图 14.9 互连线电路的阶跃响应

正如在 $v_L(t)$ 图中看到的，寄生参数的影响使上升变慢。应当看到最重要的一点是在负载门把它的输入解释成逻辑 1 值之前，$v_L(t)$ 必须达到逻辑 1 的电压（图中表示为"1"）。

当 $v_L(t)$ 在时间 t_1 从 V_{DD} 过渡到 0 V 时，发生相反的情形。当用下式模拟驱动电压时：

$$v(t) = V_{DD}[1-u(t-t_1)] \tag{14.25}$$

得到如下形式的衰减电压响应：

$$v_L(t) = V_{DD}e^{-(t-t_1)/\tau} \tag{14.26}$$

上式在 $t\geqslant t_1$ 时成立。$v_L(t)$ 的下降值必须达到 0 V。负载门才能感受到这个过渡。这个简单的例子说明互连线寄生参数在 VLSI 电路中总是引起信号延迟。

另一个例子显示在图 14.10 中。图 14.10(a) 中原来的电路图含有两条互连线，在它们之间有一个传输管。对这两条线和管子均采用 RC－π 模型就得到了图 14.10(b) 的等效电路。运用第 6 章介绍的方程计算 nFET 的寄生参数 R_n，C_D 和 C_S。注意在内部节点的并联电容可以合并成：

$$\begin{aligned} C_1 &= C_D + \frac{C_{line,1}}{2} \\ C_2 &= C_S + \frac{C_{line,2}}{2} \end{aligned} \tag{14.27}$$

这个简化的模型可以采用电路模拟程序来分析，以确定延时特性。

采用一个 m 级的梯形网络模拟互连线可以得到更为精确的分析。这种方法的缺点是在模拟这个电路时所需要的计算时间将增加。例如，SPICE 分析一个 n 个节点电路所要求的 CPU 时间随 n^2 增加。即便采用简单的互连线模型，大的 VLSI 电路部分也需要几个小时的模拟时间。通常比较有效的方法是对每根连线提供一个延时模型。然后运用这些模型作为电

路模拟中的宏单元。

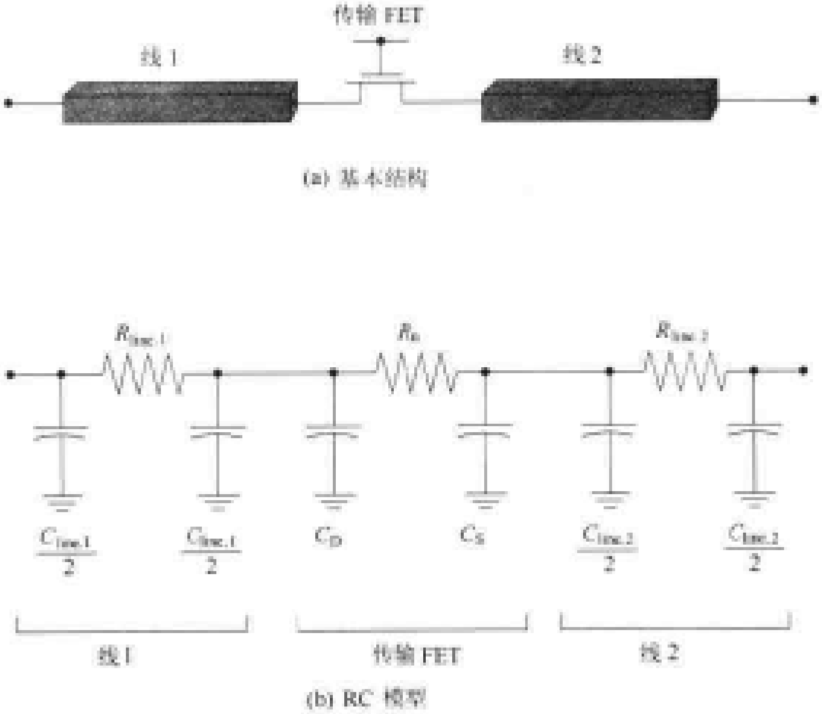


图 14.10　串联有传输 FET 的互连线模型

虽然 RC 梯形网络易于看明白，但对图 14.5 中电压 $v(z,t)$ 的建模问题从根本来说是微分性质的。在这一层次上对互连线的分析得到了如下的偏微分方程：

$$\frac{\partial^2 v(z,t)}{\partial z^2} = rc\frac{\partial v(z,t)}{\partial t} \tag{14.28}$$

上式把这一电压描述成位置和时间的函数。当阶跃输入电压即

$$v(z=0,t) = V_{DD}u(t) \tag{14.29}$$

作为边界条件时，在一条无限长线上的电压为：

$$v(z,t) = V_{DD}\mathrm{erfc}\left(\sqrt{\frac{rc}{4t}}\,z\right)u(t) \tag{14.30}$$

式中 $\mathrm{erfc}(\xi)$ 是**互补误差函数**，一般地 $\mathrm{erfc}(\xi)$ 随变量 ξ 的增加而减少，并且可用一个积分式来描述：

$$\mathrm{erfc}(\xi) = \frac{1}{\sqrt{\pi}}\int_{\xi} e^{-\alpha^2}\,d\alpha \tag{14.31}$$

对它微分得到斜率为

$$\frac{d}{d\xi}([\mathrm{erfc}(\xi)]) = -\frac{1}{\sqrt{\pi}}e^{-\xi^2} \tag{14.32}$$

上式具有高斯分布形式。然而注意在这一情形中 $\xi=\xi(z,t)$，所以在运动中同时包括空间

和时间变化。方程(14.28)的微分式具有与热力学扩散方程相同的形式。由于这一点,这一电压被看成沿连线向下**扩散**,其中误差函数的自变量

$$\xi = \sqrt{\frac{rc}{4t}}\, z \tag{14.33}$$

描述了这一运动。这从 ξ 保持常数中可以看出:随时间 t 的增长,为使 ξ 保持相同值,位置 z 需要按非线性(平方)关系的方式增加。

在实际上,利用像 MatLab 和 MathCad 这样的计算程序提供的数值是很容易的。虽然差分方程提供了比较精确的信号延时值,但附加了实际的约束,如具有电容节点的有限长连线会使分析变得非常复杂。只有几个问题可以得到收敛形式的解,因此数值分析十分必要。由于这一点,VLSI 设计者往往在大多数的信号路径中宁可采用较简单的 RC 模型进行初步的估计。

14.2.1 信号延时与连线长度的关系

这一分析的最重要结果之一是延迟时间常数 τ 与连线长度 l 的关系。采用以上讨论的任何一种分析技术都可以得到一致的结果。

从方程(14.11)中可以得到如下形式的时间常数的最简单估计:

$$\tau = R_{line} C_{line} \tag{14.34}$$

将式(14.3)和(14.6)式代入得到

$$\tau = Bl^2 \tag{14.35}$$

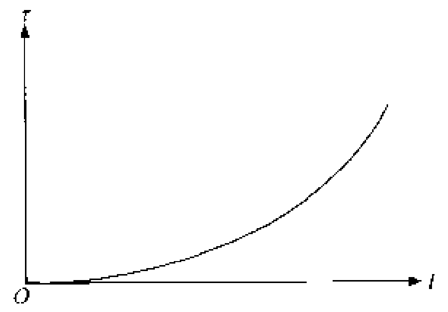


图 14.11 时间延迟与连线长度间的抛物线关系

式中 $B = rc$ 是连线的一个常数,单位为 s/cm²。这表明信号延时正比于连线长度的平方。这一平方关系显示在图 14.11 中,它对于在 VLSI 电路中采用长的互连线有重要的影响。因为不同的连线长度具有不同的信号延时,这就要求互连线必须仔细布线,特别是在关键的路径中。系统设计者必须十分仔细地精确模拟和设计互连网络以保证系统能以所希望的速度工作。

【例 14.2】 假设在长度为 50 μm 互连线上的信号延时已知为 0.13 ps。如果连线增加到 100 μm,则延迟增加为

$$\tau = \left(\frac{0.13}{50^2}\right)100^2 = 0.52 \text{ ps} \tag{14.36}$$

式中,用给出的数据计算方程中的 B。一条 200 μm 长的连线其延时为

$$\tau = \left(\frac{0.13}{50^2}\right)200^2 = 2.08 \text{ ps} \tag{14.37}$$

这表明互连线的相对长度已成为一个重要的因素。

14.2.2 对互连线延时的考虑

沿互连线的信号延时可以成为高速系统设计中的限制因素。在单独一位的关键路径中，互连线的延时必须加上正常的门延时才能得到这个问题的精确描述。它们对于全局的信号分布如在同步系统中的时钟 ϕ 变得特别重要。在以字为基础的结构中，一个 n 位字的每一位必须从一个单元传送到另一个单元，其中最慢的一位传输路径决定了整个字的数据流速度。采用仔细的字线布线技术试图使每一位的连线长度相同。

因为互连线延时从本质上来讲是电路和版图的问题，所以细节的分析通常是由电路设计组来完成的。他们的任务是建立精确的电路模型用来模拟程序而不会耗费过多的计算机时间。设计手册通常提供电路或代码形式的这类信息，它们可以直接供其他设计者使用，把数据代入参数中。高层的系统和逻辑设计者们就能估计沿所有路径的互连线延时，进行体系结构的验证。

互连线延时的重要性不能低估。在后面讨论具体问题时将会介绍更多的例子。在这些例子中，下一章讨论的全局时钟分布问题是高速同步设计中最关键的方面之一。

14.3 串扰

无论何时，当一条互连线被放在非常接近另外任何一条互连线时，这两个导体通过寄生电容就会产生耦合。如果在一条线上加上一个脉冲电压就会在所有与之耦合的连线上引起一个寄生信号。这一现象称为**串扰**。由于在一个逻辑门输入端的寄生信号可以引起一个不正确的输出，所以处理好串扰问题是设计高密度 VLSI 芯片的一个非常重要的方面。

考虑图 14.12 所示的版图，这里连线 1 和连线 2 通过寄生电容相互耦合在一起。寄生电容随这两个导体间距离的减少而增加。最强的耦合发生在当两条连线间距为最小距离 S 的时候。假设线 1 和线 2 分别具有电压 V_1 和 V_2，用 C_c 来表示总的耦合电容。电压差 $V_{12} = (V_1 - V_2)$引起从线 1 至线 2 的电流 i_{12}，它可以用如下基本的电容方程来描述：

$$i_{12} = C_c \frac{dV_{12}}{dt} = C_c \frac{d(V_1 - V_2)}{dt} \tag{14.38}$$

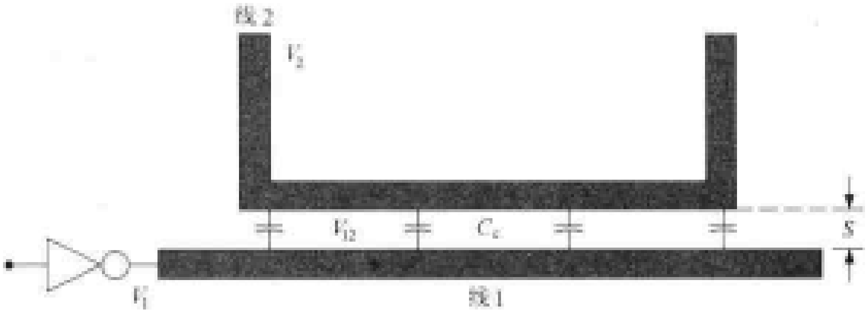


图 14.12 两条连线间的电容耦合

上式表示了电容串扰的基本含义。如果 C_c 很大，或者电压差 $(v_1 - v_2)$ 随时间变化得很快，则存在很强的耦合。因为高速设计要求有较大的时间导数（这相应于快速变化信号），VLSI 设计通常通过减少 C_c 然后考察对切换的限制来解决串扰问题。

图 14.13 为两条相距 S 的相邻互连线（标明为 1 和 2）的横截面图。每单位长度耦合电容的经验公式为[10]

$$c_c = \varepsilon_{ox}\left[0.03\left(\frac{w}{T_{ox}}\right) + 0.83\left(\frac{t}{T_{ox}}\right) - 0.07\left(\frac{t}{T_{ox}}\right)^{0.222}\right]\left(\frac{S}{T_{ox}}\right)^{-1.34} \qquad (14.39)$$

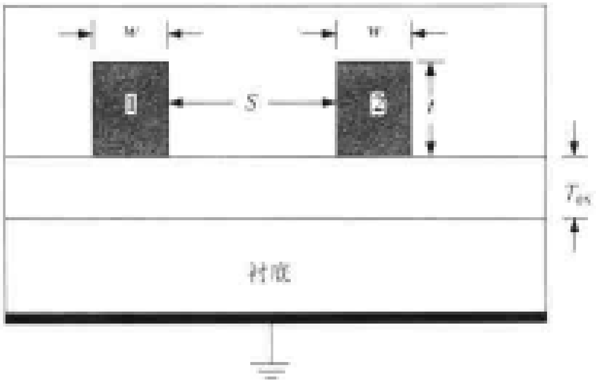


图 14.13 用于耦合电容计算的几何图

单位为 F/cm。总的耦合电容 C_c（法拉）由下式计算

$$C_c = c_c l_c \qquad (14.40)$$

式中 l_c 是间距为 S 的耦合部分的长度。这表明 C_c 是随 $(1/S)^{4/3}$ 增长的，所以采用较小的线间距离会增加耦合电容。版图设计规则考虑了这一事实，它规定了 S_{min} 的数值。基于串扰的设计规则比起光刻能够制造的最小线间距离更为重要。

图 14.14 为运用耦合电容 C_c 的一个例子。原图[图 14.14(a)]及版图的尺寸提供计算寄生参数包括 C_c 所需要的细节。这可以用在如图 14.14(b)所示的一个集总元件等效模型中。这一方法采用对称的 RC 等效电路代表每条互连线，并且采用在中间的一个电容值为 C_c 的电容模拟耦合效应。图 14.15 为另一个模型，它把耦合分成两个电容 $(C_c/2)$；还可以有其他连接方式。

一条连线的总电容是由自身电容（从这条连线至地间的电容）和任何耦合电容构成的。分别用 C_{11} 和 C_{22} 来表示线 1 和 2 的自身电容。这些就是每条连线合适的 C_{line} 值。从进连线 1 看进去的总电容为：

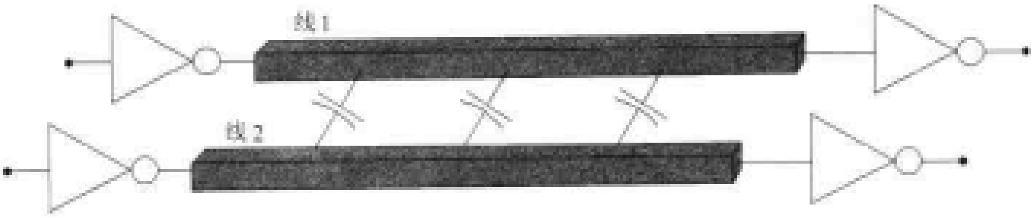
$$C_1 = C_{11} + C_c \qquad (14.41)$$

类似地，线 2 的总电容为：

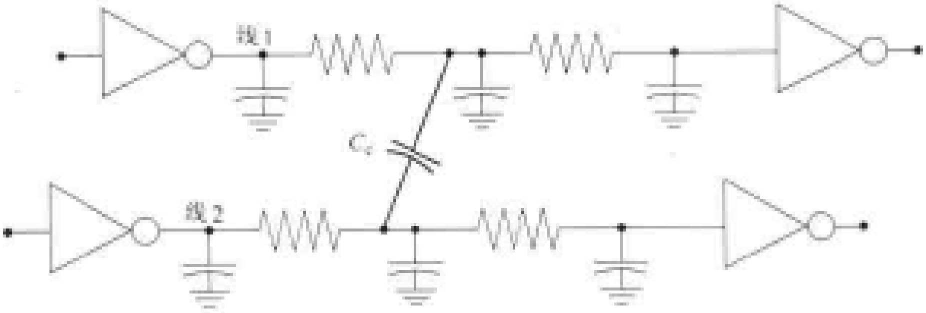
$$C_2 = C_{22} + C_c \qquad (14.42)$$

这些值对于设计每条线的驱动电路是很重要的。如果一条互连线与两条相邻的线耦合，那么这两条线都会对总电容有影响。图 14.16 显示了线 1 同时和线 2 与线 3 相互作用的情形。在相距很近的部分线 1 每单位长度的总电容为：

$$c_1 = c + 2c_c \text{ F/cm} \tag{14.43}$$



(a) 物理结构



(b) RC 模型

图 14.14　集总元件的耦合电路模型

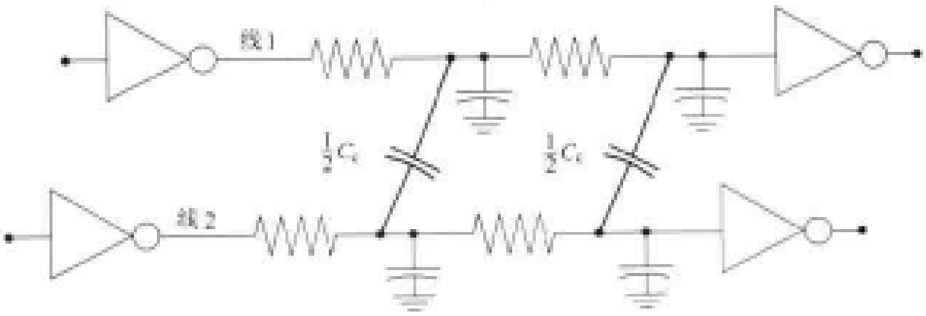


图 14.15　耦合电路的另一种模型

将它乘以长度就得到总电容。理论上在芯片中的每个导体都与其他导体相互作用。实际上，通常把自己只限制在**最近相邻**的耦合上；因为随着 S 的增加 c_c 会减少，因而这是合理的。

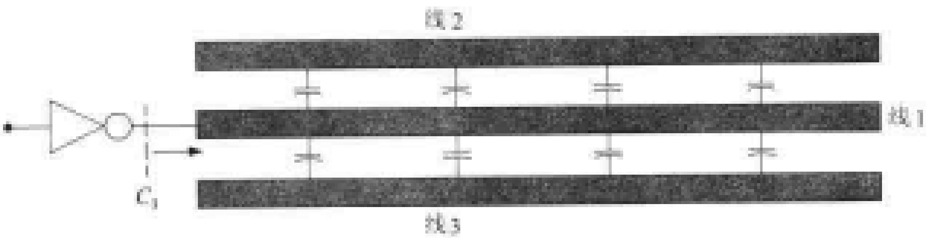


图 14.16　多条连线耦合

让我们来考察一下三条线相互作用的物理现象。对最近相邻的耦合，在线 1 上的电荷 Q_1

可写成：

$$Q_1 = C_{11}V_1 + C_{12}(V_1 - V_2) + C_{13}(V_1 - V_3) \tag{14.44}$$

式中 C_{12}和 C_{13}分别是从线 1 至线 2 和从线 1 至线 3 的耦合电容，而 v_i 是第 i 条线上的电压 ($i=1,2,3$)。在线 2 和线 3 上的电荷为：

$$\begin{aligned} Q_2 &= C_{21}(V_2 - V_1) + C_{22}V_2 \\ Q_3 &= C_{31}(V_3 - V_1) + C_{33}V_3 \end{aligned} \tag{14.45}$$

因为它们之间没有相互作用，这些方程可以联合起来得到如下的矩阵形式：

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} (C_{11}+C_{12}+C_{13}) & -C_{12} & -C_{13} \\ -C_{21} & (C_{22}+C_{21}) & 0 \\ -C_{31} & 0 & (C_{33}+C_{31}) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{14.46}$$

可以看到，这个电容矩阵是对称的，即 $C_{ij} = C_{ji}$。由于电流是电荷对时间的导数，计算：

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} (C_{11}+C_{12}+C_{13}) & -C_{12} & -C_{13} \\ -C_{21} & (C_{22}+C_{21}) & 0 \\ -C_{31} & 0 & (C_{33}+C_{31}) \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{14.47}$$

这表明在线电压上的任何变化($\mathrm{d}V_1/\mathrm{d}t$)会同时改变 $i_2(t)$和 $i_3(t)$，其影响的程度取决于电容的大小和电压变化率。类似地，改变电压($\mathrm{d}V_2/\mathrm{d}t$)或($\mathrm{d}V_3/\mathrm{d}t$)引起 $i_1(t)$电流。图 14.17 为这个三连线网络的电路级模型。它可以用标准的电路模拟器来分析。

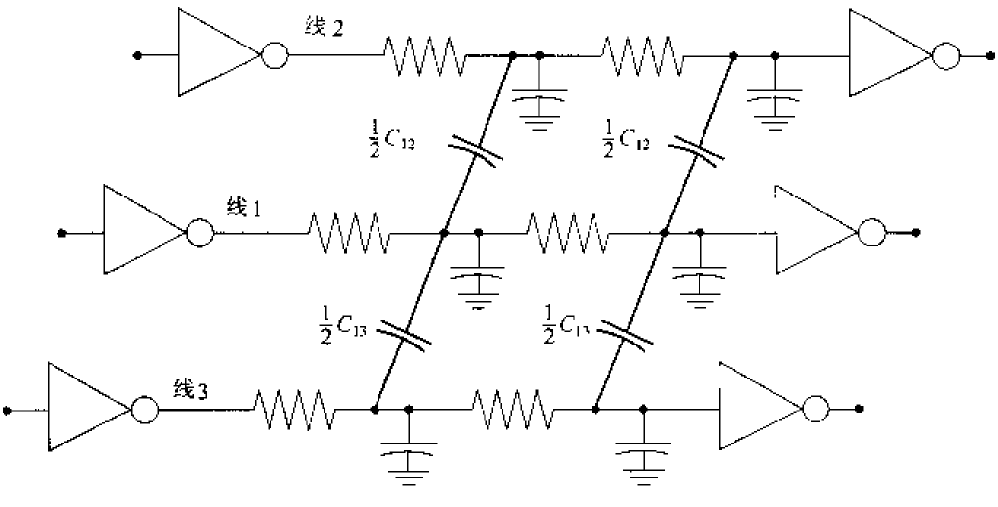


图 14.17 三线耦合问题的电路模型

至此，值得介绍一个如图 14.18 所示的绝缘平板电容公式。总的平板电容 C_p[法拉]可以用下式来估计

$$C_p = \varepsilon_{ox}\left[1.15\left(\frac{A}{T_{ox}}\right) + 1.40P\left(\frac{t}{T_{ox}}\right)^{0.222} + 4.12T_{ox}\left(\frac{t}{T_{ox}}\right)^{0.728}\right] \tag{14.48}$$

式中 $A=wl$ 是平板的底面积，而 $P=2(w+l)$ 是周长。在这一表达式中，第一项考虑了底部及其边缘的影响，而第二项加上了侧面的影响，最后一项考虑了四角的影响。

串扰也发生在不同材料层的上下重叠的连线之间。图 14.19 为 Metal 2 越过 Metal 1 线的情形。关键的参数是在这两层之间的氧化层厚度 $T_{ox,12}$。对重叠电容 $C_{ov,12}$ 最简单的近似可以采用如下的平行板公式：

$$C_{ov,12} = \frac{\varepsilon_{ox} A_{ov}}{T_{ox,12}} \tag{14.49}$$

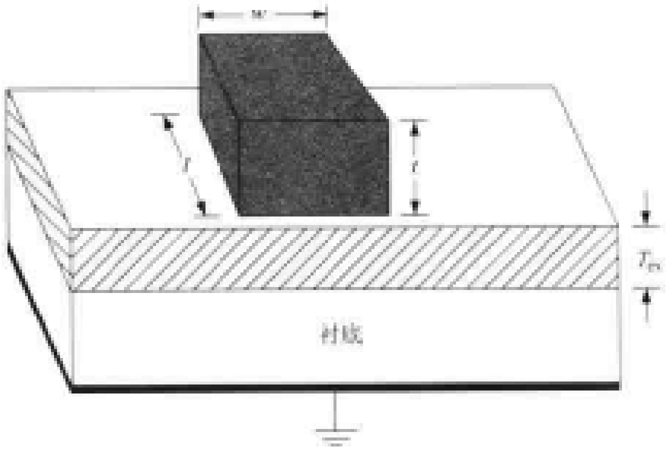


图 14.18 平板电容的几何图形

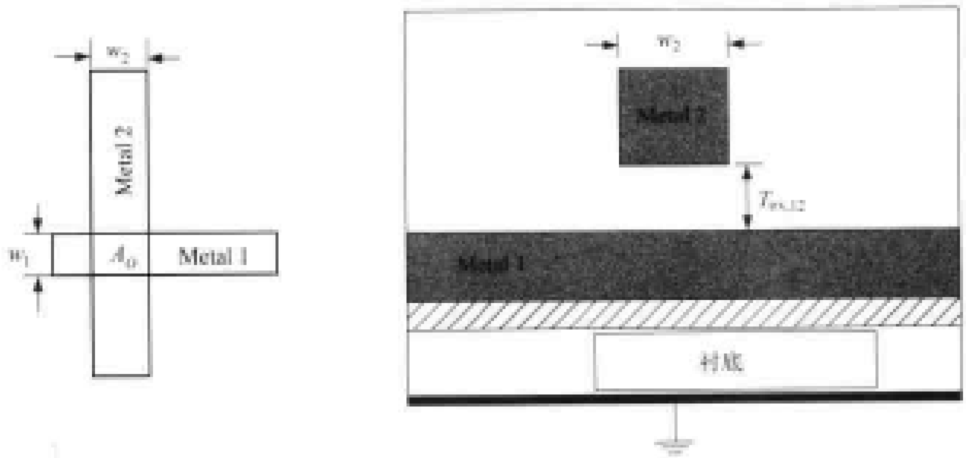


图 14.19 重叠电容

式中 $A_{ov}=w_1w_2$ 是重叠面积。尽管它忽略了边缘电场，但它对小的重叠面积的计算是足够的。由于希望在电路中所有的电容减到最小，所以画版图的技巧是把给定层上的连线布置成与它直接上面或下面一层中的连线相垂直。换言之，力图把 Metal 1 线布置成垂直于 Metal 2 线，Metal 2 线垂直于 Metal 3 线，等等。

有关串扰的考虑

串扰问题是非常复杂的并且常常要求专门的一组人员来研究。虽然可用简化的方程来估

计耦合参数，但已经开发计算机程序直接通过麦克斯韦尔电磁场理论方程来计算二维和三维的耦合参数。除了有关场强及其梯度的详细信息而外，这些程序还提供像 c 和 c_c 这样的参数值可以直接用于计算电容。连线电阻和电容用来建立等效电路的模型，然后运用像 SPICE 这样的程序进行模拟研究。

详细考察串扰通常是电路设计者和电磁学专家的领域。VLSI 系统设计者们通常可用各种参数形式表示的研究结果，如在节点上噪声的波动水平。另外，这些研究结果也会在器件和系统层次上更改设计规则。

14.4 互连线的尺寸缩小

尽管尺寸缩小原理② 最初用来描述 FET，但它也能应用于互连线并且得到有用的结果。这与如下的看法是一致的：即缩小管子尺寸是与缩小互连线尺寸一起来影响一个电路的总体性能。

一条互连线在工艺中要设定的三个几何尺寸是宽度 w，厚度 t 以及氧化层厚度 T_{ox}。改进后的光刻可以使我们把线宽减至较小的值：

$$\tilde{w} = \frac{w}{s} \tag{14.50}$$

式中 $s>1$ 是缩小因子。这是缩小一条互连线表面几何尺寸的最基本影响。

为了理解它是如何影响电性能的，回想一下，一材料层的薄层电阻为

$$R_s = \frac{\rho}{t} \tag{14.51}$$

电阻率并不因为缩小 w 而变化，所以每单位长度的连线电阻会增加，这可以从下式看出：

$$\tilde{r} = \frac{R_s}{\tilde{w}} = sr \tag{14.52}$$

假设连线长度 l 按下式缩短：

$$\tilde{l} = \frac{l}{s} \tag{14.53}$$

那么总的连线电阻不变，因为③

$$\tilde{R}_{\text{line}} = \tilde{r}\tilde{l} = R_{\text{line}} \tag{14.54}$$

每单位长度的电容随表面尺寸的缩小而减少，这只要注意下式中第一项减小就可以看出：

$$\tilde{c} = \varepsilon_{ox}\left[1.15\left(\frac{\tilde{w}}{T_{ox}}\right) + 2.8\left(\frac{t}{T_{ox}}\right)^{0.222}\right] \tag{14.55}$$

由此得到

② 尺寸缩小原理曾在 6.5.1 节中介绍过。

③ 注意缩小一个版图并不意味互连线长度按相同的方式缩短。

$$\tilde{c} = \approx \varepsilon_{ox}\left[1.15\left(\frac{\tilde{w}}{sT_{ox}}\right) + 2.8\left(\frac{t}{T_{ox}}\right)^{0.222}\right] \tag{14.56}$$

如果可以忽略边缘效应或者假设第一项占主要地位，则

$$\tilde{c} \approx \frac{c}{s} \tag{14.57}$$

因此缩短连线长度 l 可以得到新的连线电容的近似值如下：

$$\tilde{C}_{line} = \tilde{c}\tilde{l} = \frac{C_{line}}{s^2} \tag{14.58}$$

上式表明它按 $1/s^2$ 缩小。

一条多晶线在工艺中即使进行硅化也显示出最大的薄层电阻。在这种情形中减小连线长度以便不增加 R_{line}的值是很重要的。连线的时间常数按下式缩小：

$$\tilde{\tau} = \tilde{R}_{line}\tilde{C}_{line} = \frac{\tau}{s^2} \tag{14.59}$$

这是由于连线电容缩小的结果。如果 l 不缩短，那么 τ 就不会受到表面尺寸缩小的影响。这一结论对于时间常数主要由连线电容决定的一条任意的金属连线也是成立的。

让我们考察垂直尺寸 t 和 T_{ox}减小的情形，即

$$\tilde{t} < t \quad , \qquad \tilde{T}_{ox} < T_{ox} \tag{14.60}$$

上式对于任何垂直方向的缩小因子 $s_v > 1$ 一般均成立。减小厚度 t 会产生不希望的影响，即增加了薄层电阻，因为

$$R_s \approx \frac{\rho}{t} \tag{14.61}$$

式中电阻率 ρ 是一个常数。类似地，较薄的氧化层也增加了 c，所以无论是 R_{line}还是 C_{line}都会增加，造成了较长的延时。如果反过来增加 t 和 T_{ox}，那么 r 和 c 都会变得较小。

作为最后一种情形，让我们考察尺寸缩小如何影响耦合电容从面影响串扰的。盲目地减小相邻导线的表面几何尺寸会导致

$$\tilde{S} = \frac{S}{s} \tag{14.62}$$

式中 S 是连线的间距，如前面的图 14.13 所示。为了理解它是如何影响耦合的，让我们考察单位长度耦合电容的基本公式

$$c_c = \varepsilon_{ox}\left[0.03\left(\frac{\tilde{w}}{T_{ox}}\right) + 0.83\left(\frac{t}{T_{ox}}\right) - 0.07\left(\frac{t}{T_{ox}}\right)^{0.222}\right]\left(\frac{S}{T_{ox}}\right)^{-1.34} \tag{14.63}$$

总的变化因子

$$c_c \propto \left(\frac{1}{S}\right)^{1.34} \tag{14.64}$$

表明减小 S 会增加耦合电容。虽则实际耦合的增加也许会因其他项如 w 和 T_{ox}减小而抵消掉一些，但减少串扰比起所有其他考虑因素包括所占芯片面积来常常占主要地位。随着工艺进步使 S 值减少是可能的，但线间距离并不能像 FET 尺寸那样缩小那么多。

这一对互连线缩小的简短讨论表明如何运用理论来考虑改进性能。就它本身而言,这还是一个高度理想化的方法,由于工艺的限制,它不能在实际中实现。然而它确实可以作为对未来改进的一种促进,部分解释了为什么它仍然被认为是值得学习的。

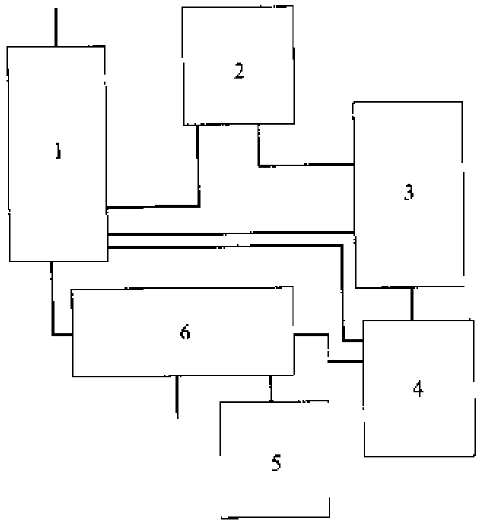
14.5 布局布线

基于单元的 VLSI 设计应用预先设计的电子电路模块,它们按需要例举以构成系统。在芯片层次上,每个模块被看成一个占据面积的块,并且必须用导线连成电路。这一步把系统和子系统的体系结构直接与硅片的物理设计相联系。在这个方面,物理设计问题与在晶体管和逻辑门版图中遇到的问题有很大的不同。长的互连线、复杂的布线网以及其他由于规模大带来的影响对于所完成设计的总体性能是非常关键的。**设计自动化**的许多特点就是解决这些问题的。

布局就是把逻辑块布置在整个设计中。这是在设计周期的早期完成的以分配面积资源,并且可以估计芯片的整体尺寸。最初的布局可以基于大的复杂的功能块,以及它们如何按照系统结构用导线连接在一起。一旦分配了面积,构成大单元的子系统设计本身就被约束了。布局甚至是在物理设计开始之前就已画好,所以它要求一组有经验的设计者基于以前的设计提供指导。

当一个逻辑模块被放入设计中时,它必须与其他的单元连接。虽然从点到点的简单的连线也许很容易加上,但现代的 VLSI 系统要求成千上百万的连接。已经开发互连线的布线技术提供结构化的方法来解决这个问题。**布局布线**的 CAD 工具对于复杂系统的布线是很有用的。设计者说明一条互连线的起点和终点,这一工具就可以产生一种布线而不会违反任何设计规则。这些程序基于不同类型的图形算法,并且都有不同程度的成功之处。

让我们首先考察布局问题。任何数字系统都可以用一种特定的方式分解成一组可以用导线连在一起的单元。图 14.20(a)为一个简单的例子。互连线表明了在不同的功能块之间的通信并且每条都具有不同的位数。如果这些功能块的尺寸是根据它们的实际尺寸按比例画出的,那么可以运用方块图来建立一个初始的布局如图 14.20(b)所示。在相邻块之间提供布线通道以利于布线。这对于使互连线长度减到最小是很重要的,并且如果局限于只有 1 或 2 层互连线的话,这一点也许是必须的。



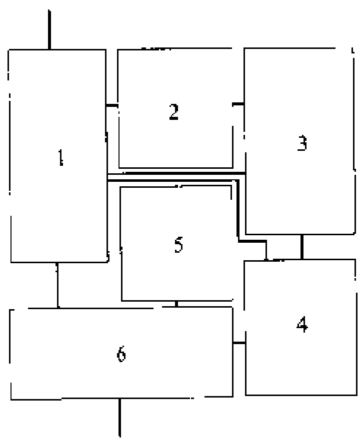


图 14.20 用方块图进行初始布局

这个例子可以用来说明可**分块(sliceble)的布局**,这是设计规模大的版图的最简单的方法之一。一个可分块的布局定义为单个模块,或者是这个布局可以采用垂直或水平线把它分隔成模块(或模块组),使这些线能够在相邻的模块组间穿过。让我们把图 14.20(b)的布局重画成图 14.21(a)的等效表示。一条垂直划分的线可以用来得到如图 14.21(b)中所示的第一次分割。图 14.21(c)的第二次分组是用两条水平划分线来得到的。这一过程可以继续进行下去直到只存在单个的模块。这个划分可以用图 14.22 所画的树结构来描述。图中的数字表示相连的模块组,而分块的过程可从每个分支下面标出的划分层次看出。注意图中最低的实体是基本的模块。一个不可划分成块布局的例子显示在图 14.23(a)中。它是不可能用水平或垂直的分隔线进行划分且不会割开任何一个模块。然而它可以作为较大的可分块布局中的一个模块,如图 14.23(b)所示。可分块的设计可以用一个树网络来描述。如果在一个可分块的设计中有一个不可分块的模块,那么树结构仍然能建立,但分割线必须以一个比较受限制的方式来定义。如果一个布局树能够建立起来,那么这个设计就称为**层次化定义的布局**。这些在概念上是很容易理解的,且已用来作为根据树结构推导版图算法的基础。

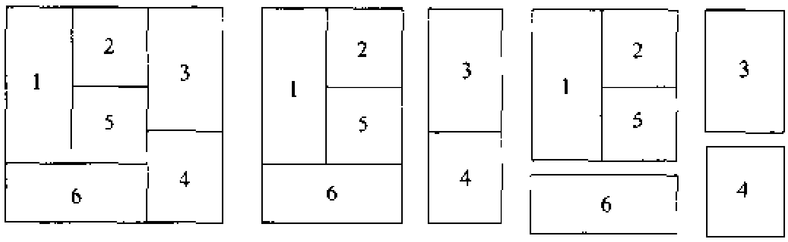


图 14.21　可分块的布局例子

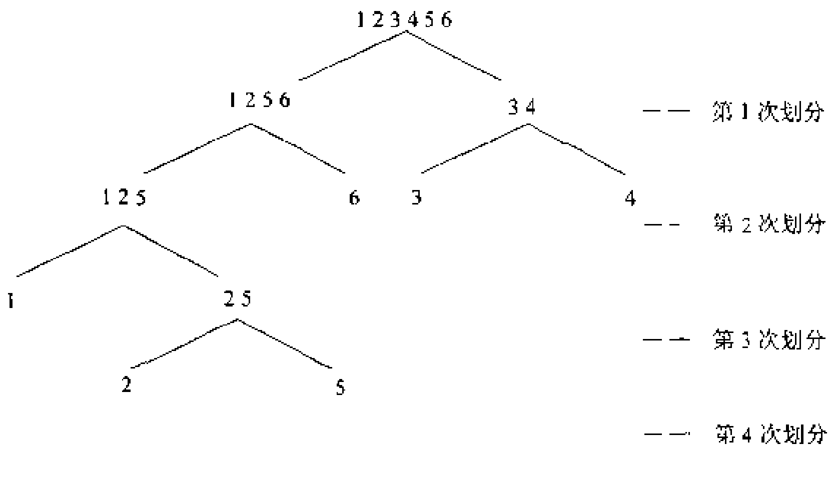
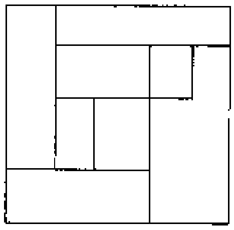


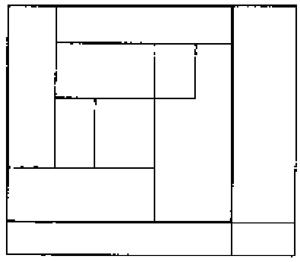
图 14.22　二进制划分树

布局的一种方法是采用**模拟退火**的概念,它基于在固态材料中形成晶体的方式。回想一下,注入硅圆片的杂质原子引起了晶体结构的破坏,它而后采用退火步骤来恢复。当圆片被加热时,把热能给了原子且引起运动。当温度降下来之后,原子寻找最小能量位置使它们放在晶格中所要求的点上。模拟的布局退火是以二进制树顶点的可能组合为基础的算法技术;这个

方法对于树结构的依赖性使它局限于可分块的布局。



(a) 不可分块布局



(b) 作为可分块设计的一部分

图 14.23 不可分块布局的例子

考虑一个描写复杂系统的可分块布局。通常有可能建立一个以上的二进制树来描述这个布局。由于每个树都是合法的，所以多个树的存在扩展了这个问题的**答案空间**。模拟退火利用了多个答案的优点，即它顺序考察这个问题不同的组合。在该树的每个顶点上比较各种不同的答案，并且选择其中最好的一个。然后移向下一个顶点使这一过程重新开始。所以整个答案是以一步一步的方式形成的。每种选择都有一个在每个顶点达到最小的**代价指标**。这个步骤得到一个**全局代价函数**，它可以用来评估这个答案。

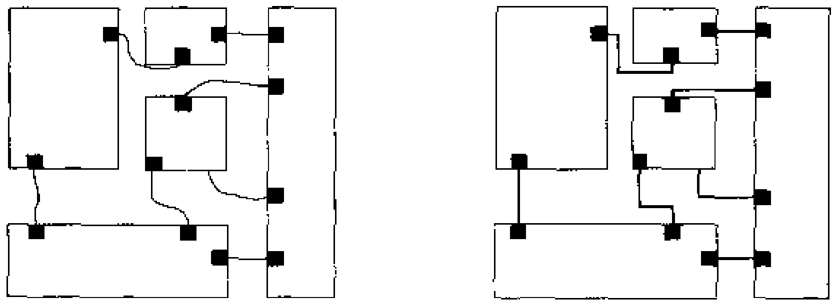
布局的其他算法有类似的特点。每个往往集中在运用几个定义得很好的约束条件和最小化的参数来设计布局。重复进行这些过程可以对许多答案进行比较。基于层次化的方法已经证明在许多情形中是有用的，线性规划软件也已用来解决这个问题。

布局的一个根本问题是定义模块本身。模块的数目和布线的网点构成布局树的结构，而它又反过来影响设计。从 VLSI 设计层次的研究中得知，基于单元的设计产生了许多不同层次的复杂性和实现方法。给定一个随意的数字模块，它总可以分解成较小的单元，或者用来作为在一个较大模块中的基本单元。如果一个模块只有一种实现，那么它称之为一个**固定单元**。常常可以采用两个或更多个不同的单元设计，这就是所谓的**可变单元**的概念。一个自上而下的布局算法在模块的特性改变时就会产生不同的结果。当采用可变单元时，这个问题就变得很复杂，因为每个设计将具有不同值的芯片面积、功耗水平、信号与互连线延时以及其他关键的参数。**布局尺寸**问题在近年来受到相当的注意，这部分是由于较大芯片不断增加了版图的复杂性。在大多数方法中一个通常的约束是要求矩形的单元，因为一般来说，它们最易用来解决布局和单元布置的一般问题。

一旦总的布局建立起来，互连布线成为一个关键考虑的问题。尽管布线会在选择布局时考虑，但互连引起的延时和时序问题直到物理设计很好进行之前是不明白的。布线通常分两步进行：全局的和细节的。全局布线涉及发现系统层次上的连接路径而不去说明实际的几何信息。细节布线运用全局的结果并且提供版图的细节，如所用的层次以及通孔的布置等。图 14.24 为这两种布线的概念。一旦细节布线完成，互连线延时就可确定，然后设计必须经过验证。

全局布线通常采用图形理论来建模。许多不同的布线模型已在文献中发表，有些比其他更为直观一些。布线模型必须与工艺以及版图设计风格有密切的联系，其中最简单的是栅格模型和以棋盘格为基础的算法。栅格模型的一个例子显示在图 14.25 中。栅格点分布在布局

图上；在模块之间的栅格点用来连线以构成要求的网点。**连线搜索**算法提供从起始点到终点的布线路径，它产生这两点间不为模块所阻的布线路径。当源点表列出的一条线与终点表列出的一条线相交时，就找到了一个答案，如在图中所示的从 A 到 B 的路径。如果找不到一个答案，那么就建立起附加的表格并链接到源点和终点的数据。另一种技术是**迷宫布线**方法，它首先考察来自源点的可能路径，然后向外扩展直到终点被一条路径碰上。图中画出的从 C 至 D 的路径则是运用迷宫概念布线的一个例子。已经努力研究了许多发现路径的算法以找到快速和有效的布线方法。



(a) 全局　(b) 细节

图 14.24　布线步骤

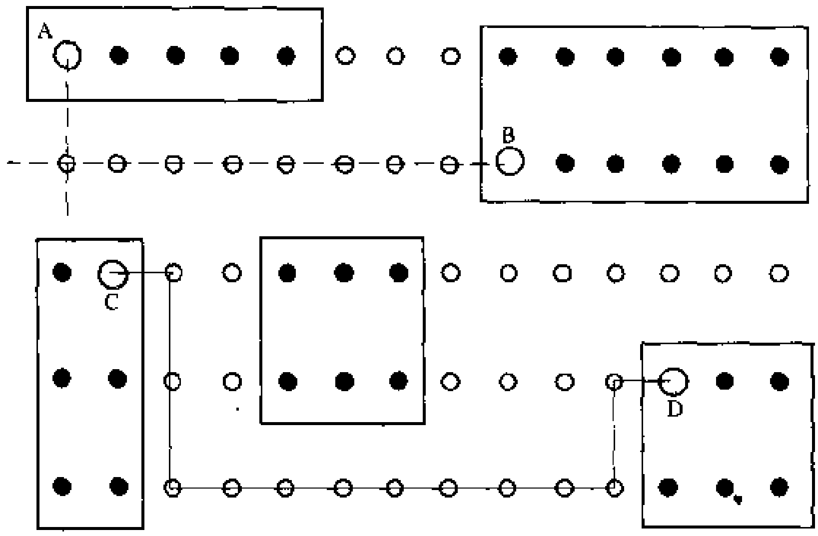


图 14.25　全局布线的栅格模型

细节布线则是每次解决全局解答中的一个区域。布线区域分成两类。**通道**是一个矩形布线区域，它的两边以模块为界；一个**开关盒**区域则是四边都有边界的矩形区域。给定一个布线区域，其解答必须能使一个版图在实际上可以放入这个被分配的区域中去。增加互连金属层的数目使布线比较容易，但布置通孔却成为问题。

设计自动化领域发展成为 VLSI 中一个非常成熟的学科。解决这些复杂性和验证问题的新产品不断出现。随着芯片密度和复杂性的继续增加，开发更好和更有效的 CAD 工具对于推进这一学科的发展已必不可少。

14.6 输入和输出电路

芯片上的CMOS电路通过封装与外部世界相隔离,这相当程度地简化了设计环境。电容的大小被限制在fF(10^{-15}法拉)的水平上,且只存在局部的相互作用。然而在某些点上,电路必须与使用该芯片的外部世界相互作用。普通的印刷电路(PC)板的电容可以很容易地达到50~100皮法拉(pF)的范围。静电荷可以累积至超过50 kV的电平,它高到足以破坏灵敏的MOS管。因此必须设计输入和输出电路来隔离这些内部电路,以避免受到这类问题的影响。

在VLSI电路和外部世界之间的接口可以分为三类:输入、输出或双向接口。这些显示为图14.26中的压焊块I/O。输入电路设计用来保护敏感的CMOS电路免受高电压的影响,而输出电路的设计则往往集中在驱动高电容的印刷板连线。双向电路则是这两者的联合,它使一个端口既可作为输入又可作为输出。

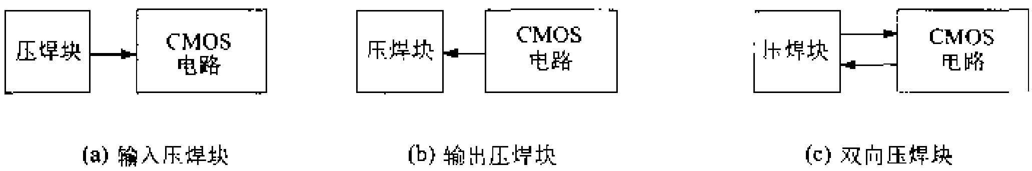


图14.26 输入/输出口的类型

14.6.1 输入电路

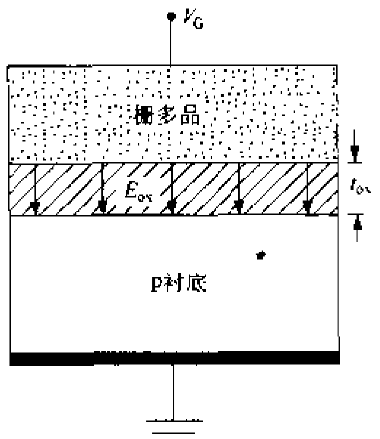


图14.27 MOS结构中的氧化层电场

静电荷往往累积在电介质(非导体)的表面且能引起极高的电压。引起静电荷的一个通常的原因是摩擦的相互作用。[④] 许多人在这方面已有过这样或那样的经历。走过一个铺地毯的房间可以累积起足够的电荷,在触摸门把手时会产生火花。事实上,这个电压已上升到足够大的数值使空气的绝缘被击穿。

大多数场效应管对于静电放电(ESD)现象极为敏感。考虑图14.27所示的MOS结构,图中栅氧的厚度t_{ox}一般小于100Å=10 nm。当加上栅电压V_G时氧化层的电场E_{ox}(V/cm)可以估计为:

$$E_{ox} \approx \frac{V_G}{t_{ox}} \tag{14.65}$$

式中$V_G=3$ V,$t_{ox}=0.01$ μm。在绝缘层中的电场值约为$E_{ox} \approx 3\times10^6$ V/cm。它取决于工艺,能够加在一个二氧化硅绝缘层上的最大电场的典型值约在$(5\sim10)\times10^6$ V/cm的数量级上。如果

④ 摩擦研究是摩擦学领域的基础。

电场超过了这个值，则发生击穿且有电流流入衬底。这会破坏氧化层的绝缘特性并且因此破坏管子特性。甚至一个坏的 FET 也会使整个芯片毫无用处，因此这个问题必须严肃对待。

静电荷会在整个制造过程中产生。此外，芯片每天还必定会经历一些过程如运送、卸货和插入板中。考虑到这一点，现代的 CMOS 芯片都设计有电路来保护输入管免受过多静电荷的影响，而且这些器件的运送采用导电泡沫(foam)。即便采用这些安全措施，在对待这些芯片时仍然必须小心遵循制造商的建议。即采用接地工作平面和手腕带，减少静电荷的数量，因为它们能把过多的电荷导入地中。

CMOS 输入保护电路是用来提供从管子放电的路径。这一电路必须对输入信号比较透明，只有在外加不正常的电压时才起作用。最通常的设计是采用电阻－二极管电路来提供电荷衰减的路径。图 14.28 总结了一个 pn 结二极管结构的有关特性。一个二极管的阳极和阴极是由 p 和 n 区域定义的，如图 14.28 (a)。电压 V 定义为正时，其极性为"＋"在阳极而"－"在阴极。正电流 I 定义为流入阳极流出阴极。具有正电压和电流的一个二极管的电路符号显示在图 14.28 (b)中。图 14.28 (c)的 $I-V$ 特性显示了 I 是 V 的函数。$V>0$ 的条件定义正向偏置情况，此时有相当大的电流流过。使极性反过来，即"＋"的一边在阴极而"－"的一边在阳极，定义一个反相偏置情况，此时 $V<0$。一个反向偏置的二极管在电压较低时会阻止电流流动，但**在齐纳电压** V_Z 即如图所示的 $V=-V_z$ 时会发生击穿。V_z 的值由掺杂水平确定，它也是 pn 结的特性。pn 结的击穿不是破坏性的；如果移去电压后再加电压，二极管会以正常的方式工作。

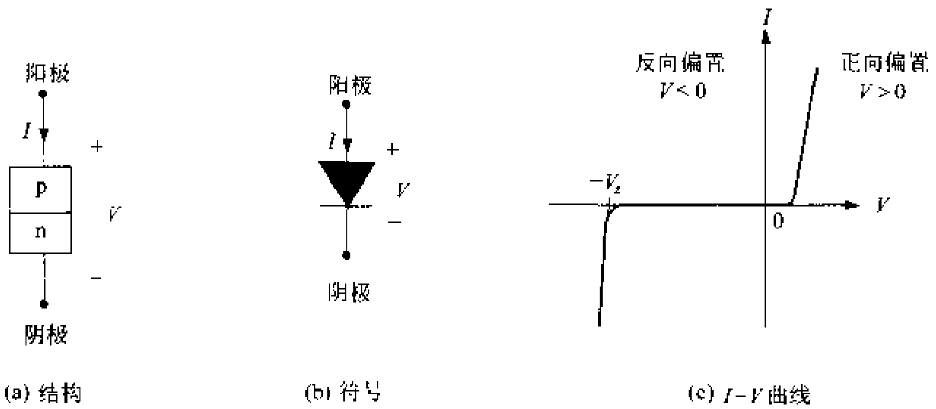


图 14.28　二极管特性

二极管和电阻一起可以用来形成图 14.29 所示的输入保护电路。如果一个过大的正电压加到输入压焊块上，那么这个电阻就会吸收沿输入线的电平。在这些情形下，二极管 D1 和 D2 发生击穿并把电荷从输入级晶体管的栅极上导走。取决于 pn 结，CMOS 电路中的这一二极管一般有 $V_z=10\sim12$ V 或更小。在物理设计层上，电阻可以用在 p 衬底上的 n 注入层来实现，如图 14.30(a)所示。n＋区域采用蛇形弯曲形状以得到一个方块版图，其电阻为：

$$R = R_{s,n}n \qquad (14.66)$$

式中 $n=(l/w)$ 是方块的数目。在这类几何形状中，一个 90°的折角不能看成一个完全的方块；静电场分析表明一个合理的估计是 $n_{corner}\approx 0.69$。反向偏置的 pn 结二极管是自动建立在图 14.30(b)所示的结构中的。无论电阻还是二极管都是分布结构而不是像在电路图中所示的分立器件。

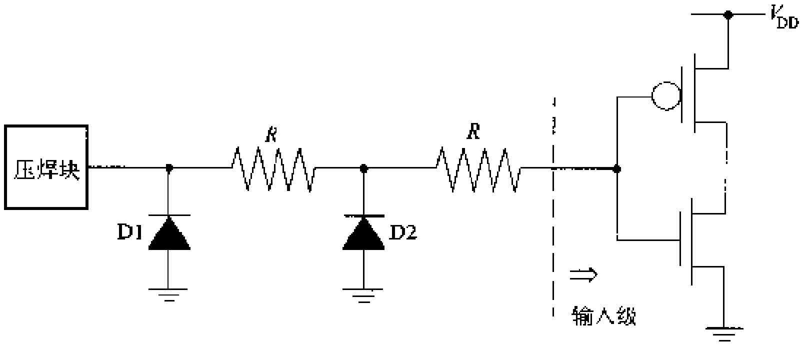
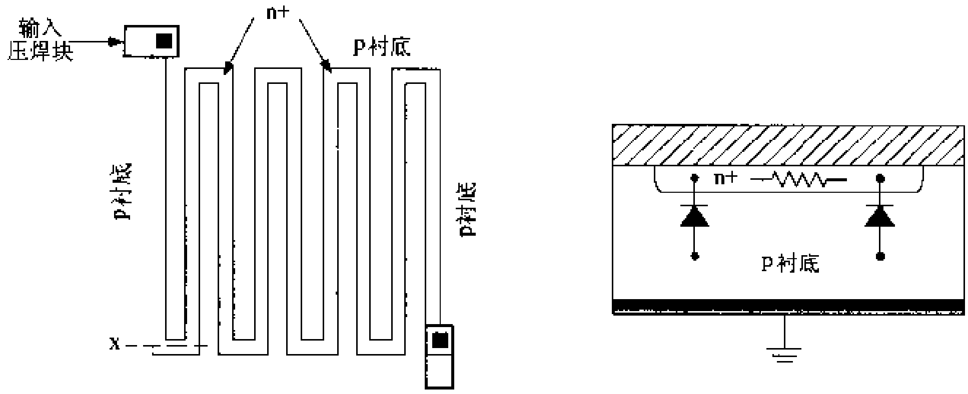


图 14.29 输入 ESD 保护电路



(a) 顶视图　　(b) ×处的截面图

图 14.30 输入电阻 二极管结构

另一种输入保护技术如图 14.31 所示。它采用二极管 D1 和 D2，但在输入和电源之间又附加了二极管 D3 和 D4。这一电路保持到达栅的 DC 电压在 $[-V_d, V_{DD}+V_d]$ 的范围内，其中 $V_d \approx 0.7$ V 是二极管的**导通电压**，也就是为产生电流所要求的电压值。电路中还包括一个高阈值电压 nFET 以提供附加的电荷吸收通路。它采用较厚的绝缘场氧(FOX)作为栅绝缘，所以较弱的场效应得到较高的阈值电压 V_{TF}，其典型值约 10～15 V。在正常工作条件下 $V_p < V_{TF}$ 因而 $i=0$。如果一个高输入电压使 V_p 增加到 V_{TF} 的值则 FET 导通，于是电流流动，使电荷离开逻辑门的输入。有些设计只采用保护 FET 及 D1、D2 二极管。

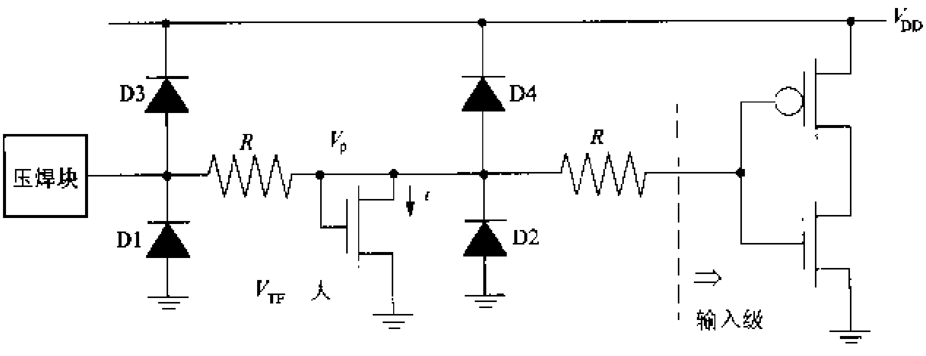
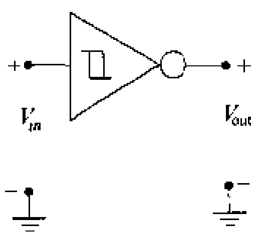
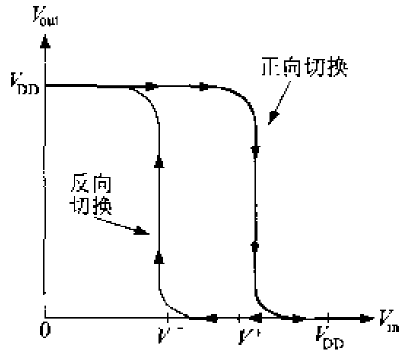


图 14.31 另一种输入保护电路

在设计输入保护电路之后，接收电路还应有其他考虑。**施密特触发器**电路常用来作为输入电路，以阻止噪声引起的虚假切换。施密特触发器的特点是在它们的电压传输曲线中有**滞环**。在电路级，滞环是指输入电压 V_{in}从 0 V 增加到 V_{DD}时所得到的输出曲线不同于 V_{in}从 V_{DD}减小到 0 V 时的曲线。图 14.32 (a)为反相施密特触发器门的电路符号；在三角形中间的图标使它区别于简单的反相器。这一图标显示图 14.32(b)中 VTC 的特征形状。当 V_{in}从 0 V 开始增加时，V_{out}保持在高电平 V_{DD}直到 V_{in}达到正向触发电压 V^+；然后 V_{out}下降到 0 V。对于反向切换，V_{in}起始于 V_{DD}然后减小，此时 $V_{out}=0$ V。输出维持在低电平直到 V_{in}下降到反向触发电压 V^-。对于 $V_{in}<V^-$，$V_{out}=V_{DD}$。注意对于一个功能正确的施密特触发器要求 $V^-<V^+$。滞环保证了在输入信号上升或下降沿上的小波动不会引起虚假的切换过程。



(a) 符号



(b) 电压传输曲线

图 14.32　反相施密特触发器

图 14.33 为 CMOS 施密特触发器电路。它采用镜像设计，nFET 决定了 V^+而 pFET 决定了 V^-的值。考虑 nFET 电路。M1 和 M2 串联并且都由输入电压驱动。当 $V_{in}=0$ 时 $V_{out}=V_{DD}$且 M3 导通。因为 M3 的漏是连在电源上的，所以它起了反馈路径的作用。当 V_{in}增加时，即使在 M1 导通之后它也保持 M2 截止。分析表明正向触发电压为

$$V^+ = \frac{V_{DD} + \sqrt{\frac{\beta_1}{\beta_3}}V_{Tn}}{1+\sqrt{\frac{\beta_1}{\beta_3}}} \quad (14.67)$$

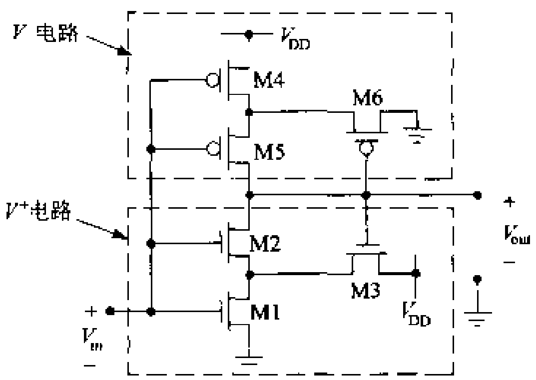


图 14.33　镜像 CMOS 施密特触发器

式中器件比(β_1/β_3)是设计变量。因为 M1 和 M3 都是 nFET，所以上式简化为器件的宽长比之比

$$\frac{\beta_1}{\beta_3} = \frac{(W/L)_1}{(W/L)_3} \quad (14.68)$$

类似地，M6 是 pFET 部分的反馈管子。反向触发电压由下式给出：

$$V^{-}=\frac{\sqrt{\frac{\beta_4}{\beta_6}}(V_{DD}-|V_{Tp}|)}{1+\sqrt{\frac{\beta_4}{\beta_6}}} \tag{14.69}$$

式中

$$\frac{\beta_4}{\beta_6}=\frac{(W/L)_4}{(W/L)_6} \tag{14.70}$$

为 pFET 比。这一电路有此特点可能使一个设计具有尺寸较大的 FET。这是因为串联管必须设计得较大才能避免电阻过大，而开关电压则是通过选择 M3 和 M6 的尺寸确定的。

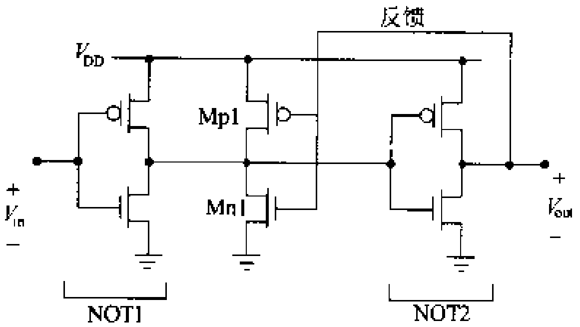


图 14.34 非反相施密特触发器电路

非反相施密特触发器电路显示在图 14.34 中。晶体管 Mp1 和 Mn1 分别作为弱的上拉和下拉器件，它们是由输出电压 V_{out} 通过反馈连接来控制的。假设 $V_{in}=0$ V，第一个反相器(NOT1)的输出为高，所以第二个反相器(NOT2)的输出 $V_{out}=0$ V。这使 Mp1 导通而 Mn1 截止。如果增加 V_{in}，NOT1 的输出节点由 Mp1 维持在高电平，它延迟了翻转。Mp1 和 Mn1 的宽长比必须较小，所以仍能允许翻转发生。

14.6.2 输出驱动器

输出电路必须驱动压焊块电容以及与引线相连的外部负载。公式(14.48)可以用来求压焊块的电容 C_{pad}，但片外负载随应用而有所不同。一个典型的设计值是约 80 pF，它大约是一个测试探针产生的负载电容。由于它比正常片内设计所遇到的飞法拉(fF)的量级大得多，所以我们必须采用大输出管来维持高速度。

【例 14.3】 假设一个 0.5 μm CMOS 工艺采用 Metal 3 层上的 I/O 压焊块，它的特征电容为 1 aF/μm。单位 aF 代表阿托法拉(attofarad)，1aF = 10^{-18}F。

如果采用尺寸为 75 μm×75 μm 的压焊块，压焊块的电容为：

$$C_{pad}=(14)(75^2)=78.75\ \text{fF} \tag{14.71}$$

它必须加到外部电容的影响中去。

片外驱动器设计是 CMOS VLSI 设计的关键点。简单逻辑门的片内切换时间是在亚纳秒的范围内。把片内高速率的数据传送到外部世界会因大的电容值而变复杂。8.3 节讨论的逐

级加大的驱动链可以用来解决这个问题。图 14.35 为一个四级输出电路，它需要驱动一个大的毫微法拉(pF)级的负载电容 C_L。理论上，使延时最小的分析得到这个链的级数 N：

$$N = \frac{\ln\left(\frac{C_L}{C_{in}}\right)}{\ln(S)} \tag{14.72}$$

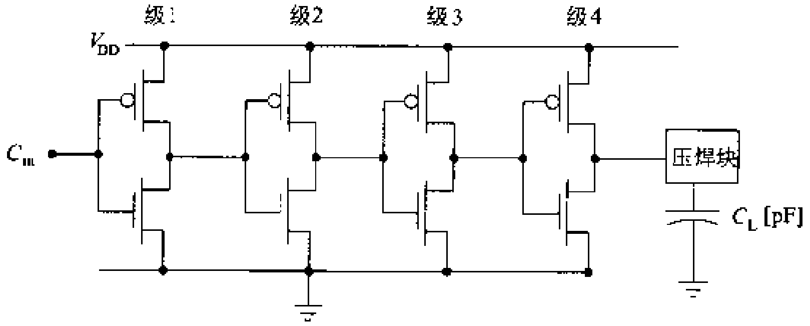


图 14.35 逐级加大的驱动器链输出电路

式中 S 是级比因子。然而大的输出电容可以导致大的 N 值和宽长比，所以比较实际的方法常常是看一下输出翻转时间的要求。在这个例子中，输出特性由第 4 级决定，因为它驱动 C_L。若用 t_r 和 t_f 来说明上升和下降时间值，那么就可以写出第 4 级的时间常数表达式

$$\begin{aligned} t_r &= 2.2R_pC_L \\ t_f &= 2.2R_nC_L \end{aligned} \tag{14.73}$$

一旦电阻已知，宽长比可以从下式计算出：

$$\begin{aligned} \left(\frac{W}{L}\right)_{p,4} &= \frac{1}{k'_p(V_{DD}-|V_{Tp}|)R_p} \\ \left(\frac{W}{L}\right)_{n,4} &= \frac{1}{k'_n(V_{DD}-V_{Tn})R_n} \end{aligned} \tag{14.74}$$

第 4 级的输入电容为

$$C_4 = C_{ox}[(WL)_{n4}+(WL)_{p4}] \tag{14.75}$$

这也就是第 3 级的输出电容。每级可以采用相同的上升和下降时间值来设计，从输出端向内部电路进行。这个过程一直重复到输入电容 C_{in} 到达一个“正常”的值，于是就决定了级数。使各级延时相等，相当于运用线性比例逐级放大。

双向压焊块同时为输入和输出信号提供电路。输入电路与上面描述的一样。输出驱动器应能进行三态操作，所以它们不会与进入的信号相干扰。图 14.36 显示了一个例子，

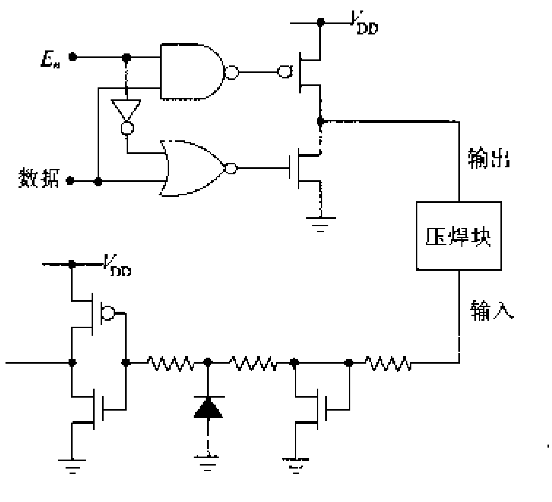


图 14.36 双向 I/O 电路

输出电路采用由 NAND2 和 NOR2 逻辑电路控制的大驱动 FET。这些门可以考虑成一个逐

级加大驱动链的一部分，因为驱动 FET 的电容很大。使能信号 En 是三态控制。当 $En=0$ 时，输出电路处于 Hi－Z 状态，压焊块可以用做输入。当 $En=1$ 时，输出电路相当于数据输入的非反相缓冲器。必须仔细确保输出信号不为输入信号所用(除非设计要求如此)。

14.7 电源的分配和功耗

电源值 V_{DD}和 V_{SS}是由外部提供的，它们通过两个不同的压焊块进入芯片内部。**电源分配网络**是一组金属线把电压送到电路的每个部分。它必须按照一定的几何形状来设计，以允许较高的集成密度而又能提供必需的电流。

两个电气问题可能在电源网格的设计中占据了主要地位。第一个是电迁移现象，即当电流密度 $J[A/cm^2]$ 很大时金属原子会从一端移动到另一端。因为总电流 I(安培)与横截面积 tw 的关系为：

$$I = J(tw) \tag{14.76}$$

可以增加导线的宽度 w 以保证 J 维持在可接受的程度。这通常是由设计规则表来说明的，它对于不同的电流范围列出了最小的宽度 w。第二个问题是导线电阻 R_{line}。按照欧姆定律，导线上的电压降为

$$V_{line} = IR_{line} \tag{14.77}$$

所以到达电路的电压被改变了这一数量。线宽、布线以及通孔的位置都会影响两点之间的总电阻。这两个问题都可以采用较宽的线来解决，但这一简单的解决办法会消耗过多的面积。

树形结构是设计分配方式最常用的方法。它的一般概念如图 14.37 所示。主干 V_{DD}线的设计是使具有足够的宽度以便传送整个电路的总电流 I。总电流送入分支中，每个分支运送一个平均电流 I_1 满足

$$I = N_1 I_1 \tag{14.78}$$

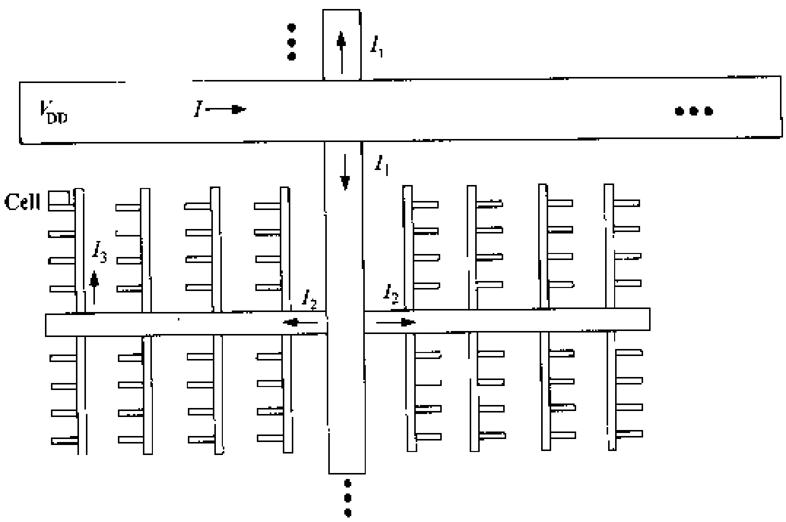


图 14.37 确定电源分配的线宽尺寸

式中 N_1 是第二级线的数目。每个二级线送到三级线，后者传送电流 I_3，依次类推，直到每个逻辑单元供上电为止。一旦电流值已知，就可以计算宽度。由于数字 CMOS 电路对电流的要求随时间变化，所以采用平均值来计算宽度。过渡特性也许要求把某些线加宽。

实际的电源网络设计是布置电源线且把它们连在一起形成一个电源网。从压焊块来的电源总线与送到单元中去的电源线通常是隔离的，以减少噪声问题。这一概念可以从图 14.38 中理解。一条信号线被放在两条 V_{SS}（接地）线之间以提供电屏蔽。从物理观点看，隔离的实现是因为耦合电容 C_c 被连到（相对）无噪声的 V_{SS}线。这一原理也可以应用到图 14.39 所示的两金属分配模型上。电源电压 V_{DD}和 V_{SS}用来形成环绕内部电路区域的电源环。V_{SS}环用来形成对单元的电屏蔽。金属线的宽度是根据逻辑电路所需要的电流来确定尺寸的。各个单元被放置在最小宽度的 V_{DD}和 V_{SS}轨道之间。

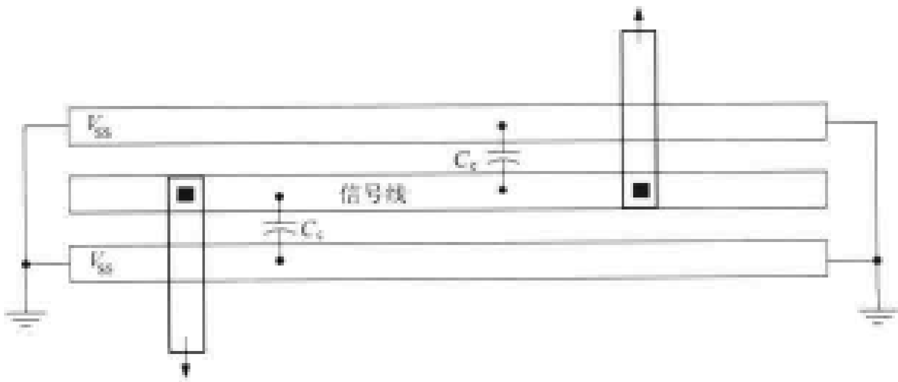


图 14.38 采用 V_{SS}线隔离

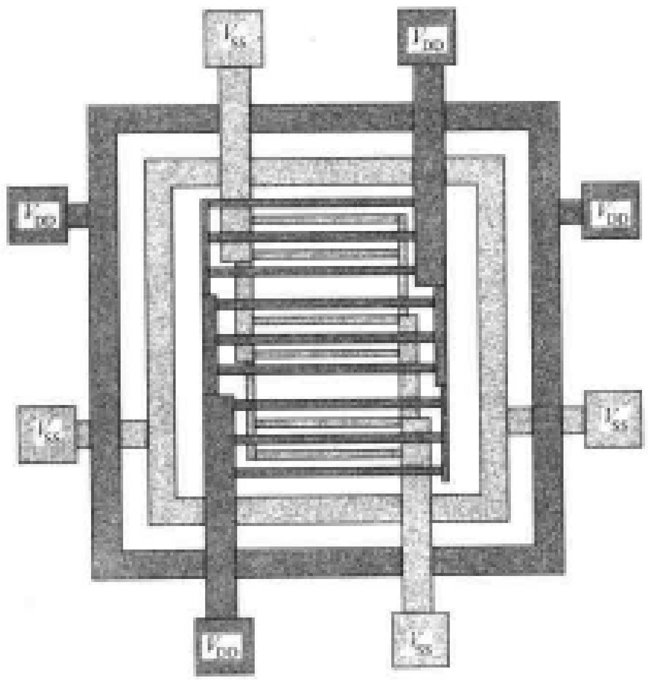


图 14.39 电源分布技术

VLSI 设计库通常具有压焊块结构单元，它可以用于芯片的电源输入，所以设计集中在内

部区域。考虑宽度和负载大小的电气设计规则在关键电路中可以按一条线一条线地来应用，但涉及的确切数字通常要求有一个基于结构和电路设计的计算方法。

同时切换引起的噪声

同时切换噪声(SSN)是许多同时切换的事件引起的连线电压的波动。这一影响在文献中也称为 **Δ－I(ΔI)和接地线抖动**。SSN 发生在当一条线上的电流随时间快速变化的时候。由于高速数字电路设计用来快速切换，所以 SSN 问题在 VLSI 中很普遍。

所有导线都具有一些**电感** L，这是由于电流的磁场能量存储特性所致。尽管通常忽略传送小电流信号的互连线中的电感效应，但它们对于供电给高速电路的电源分配线来说是十分重要的。基本的问题可以通过基本的电磁场理论来理解。一个电流 i 产生了一个磁场，它的磁通量 Φ 为

$$\Phi = Li \tag{14.79}$$

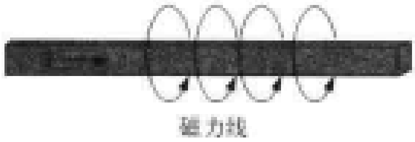
电感 L 的单位是亨利[H]并且被认为是一个比例常数。图 14.40(a)显示了它的物理模型。法拉第感应定律指出，一个时变磁通量 $\Phi(t)$ 感应出的电压 $v(t)$ 为：

$$v(t) = \frac{\mathrm{d}\Phi}{\mathrm{d}t} \tag{14.80}$$

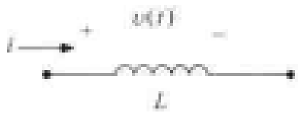
代入磁通量公式就得到了电感的 $I-V$ 关系

$$v(t) = L\frac{\mathrm{d}i}{\mathrm{d}t} \tag{14.81}$$

图 14.40(b)为电感符号。这个方程表明感应电压正比于电流的时间变化率 $\mathrm{d}i/\mathrm{d}t$。



(a) 互连线



(b) 等效元件

图 14.40　导线电感的来源

SSN 的问题通过回忆图 14.41 所示简单反相器的开关特性就可以理解。当输入电压 $V_{in}=0$ 时，nFET 截止，$i=0$；零电流情形对于 $V_{in}=V_{DD}$时也成立，因为 pFET 截止。从 V_{DD}至地的直接电流只发生在 V_{in}发生变化两个管子都导通的时候。图中画出的($\mathrm{d}i/\mathrm{d}t$)既具有正值也具有负值。如果输入电压 V_{in}可以按较快的速率变化(也即一个很陡的斜率)，则电流的导数就会增加。

现在考虑图 14.42 的情形，即一条公共的电源线供电给一组电路。如果每个门在同一时间切换，那么流过寄生电感 L_1 的电流 i_1 可以由下式计算

$$i_1(t) = \sum_j i_j(t) \tag{14.82}$$

式中求和是针对全部门的集合。电感两端的电压为

$$v_1(t) = L_1\left(\frac{\mathrm{d}i_i}{\mathrm{d}t}\right) = L_1\sum_j\left(\frac{\mathrm{d}i_j}{\mathrm{d}t}\right) \tag{14.83}$$

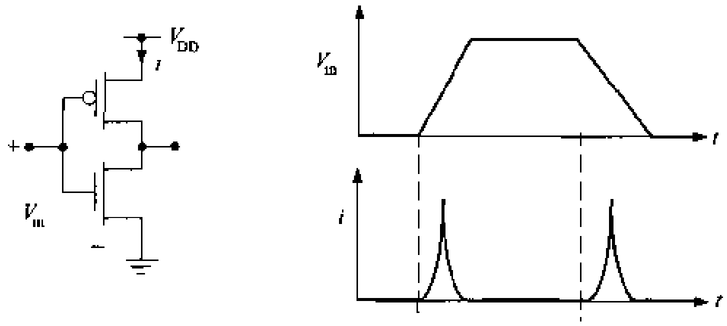


图 14.41 一个反相器电路中的电流

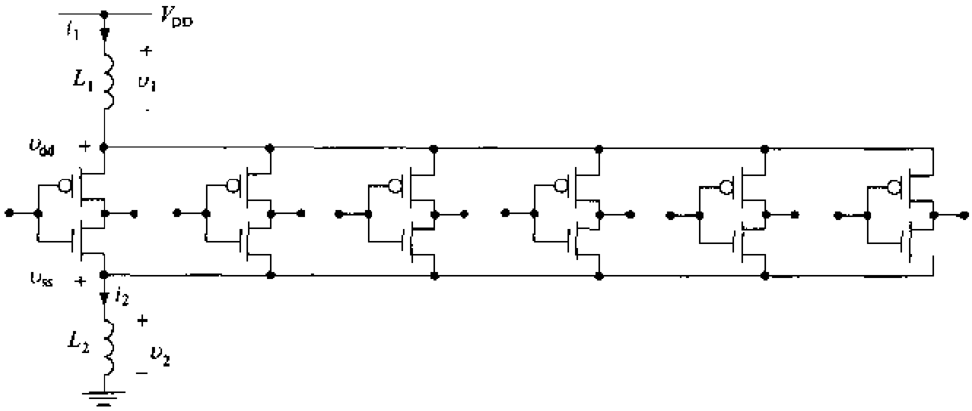


图 14.42 同时切换噪声的例子

所以加到 pFET 上电压的实际值 v_{dd} 为

$$\begin{aligned} v_{dd}(t) &= V_{DD} - v_1(t) \\ &= V_{DD} - L_1\sum_j\left(\frac{\mathrm{d}i_j}{\mathrm{d}t}\right) \end{aligned} \tag{14.84}$$

类似地 L_2 的电压 v_1 可以计算为

$$v_{ss}(t) = v_2(t) = L_2\sum_k\left(\frac{\mathrm{d}i_k}{\mathrm{d}t}\right) \tag{14.85}$$

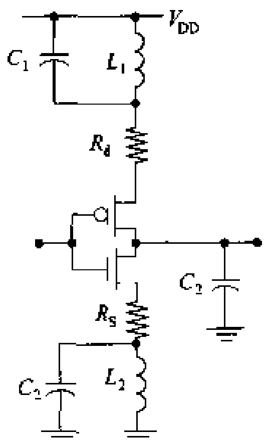


图 14.43 完整的电路模型

它是加在 nFET 源端的有效电压。一般地，i_1 和 i_2 并不等，因为它们是流入或流出门输出节点的电流。注意取决于过渡导数可正可负。一个比较完全的分析通常采用图 14.43 的电路模型。除了连线电容 C_1 和 C_2 而外，它还增加了漏和源的电阻 R_d 和 R_s 以在波形模拟中得到较高的精度。

这一分析的主要结果是 $v_{dd}(t)$ 和 $v_{ss}(t)$ 的瞬间值而不是 DC 值 V_{DD} 和 0 确定了逻辑门的输出电压。被改变的逻辑电平值可能在逻辑中引起出错。因为总电流是各个门的电流的和，

如果许多门同时切换的话,即使中等的切换速度也会引起问题。如图 14.44 所示的直方图显示在任何给定时刻发生切换的门的数目。这个数据通常是根据每个门的峰值电流来加权的。对于大量门发生变化的高翻转周期则给以特别关注。理想情况是在任何给定时刻发生变化的门的数目是一个常数,因为此时就会达到稳态的电压。

由于现代 VLSI 系统的复杂性,预见门的翻转率是很复杂的任务。而且电路设计类型和时钟对于结果有直接的影响。一个随机逻辑门串联链的最简单情形可以用图 14.45 所示的电流分布图来描述。它代表的情形是门 1 逻辑组的输入在时间 t_0 切换并且串行地传播通过这个链引起每组逻辑发生翻转。在电流尖峰之间的间隔可以被估计为门的延迟时间。这类考虑可以用在逻辑和电路设计阶段以使随时间变化的电流大小相同。

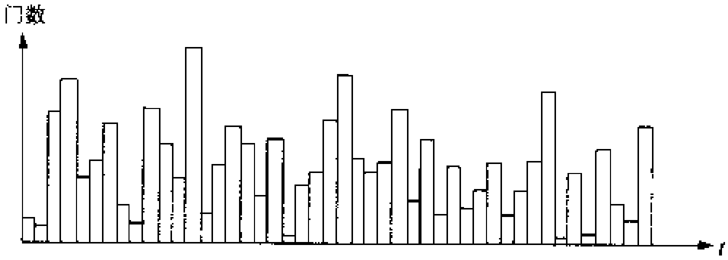


图 14.44 门的切换随时间的分布

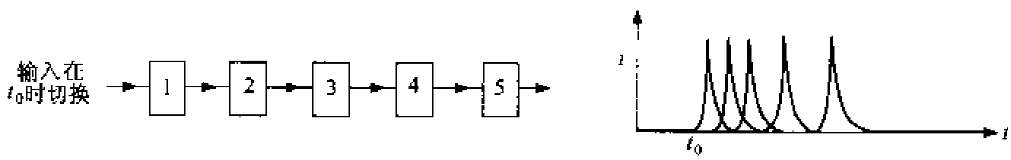


图 14.45 一条随机逻辑链中的切换电流

图 14.46 中的动态逻辑电路由于有时钟控制比较可以预见。当 $\phi = 0$ 时,电路处于预充电(P),充电电流 i_{ch}流入到输出电容 C_{out}中。当时钟变为 $\phi = 1$ 时,电路进行求值(E)。如果产生的输出为逻辑 0,那么 i_{dis}就流到地线中并且输出,将在下一个预充电期间被重新充电。如果电荷被保持,那么会发生漏电,所以这一级仍然需要部分重新充电。放电电流发生在求值阶段并且按照每级在串联逻辑链中的位置而分布开。

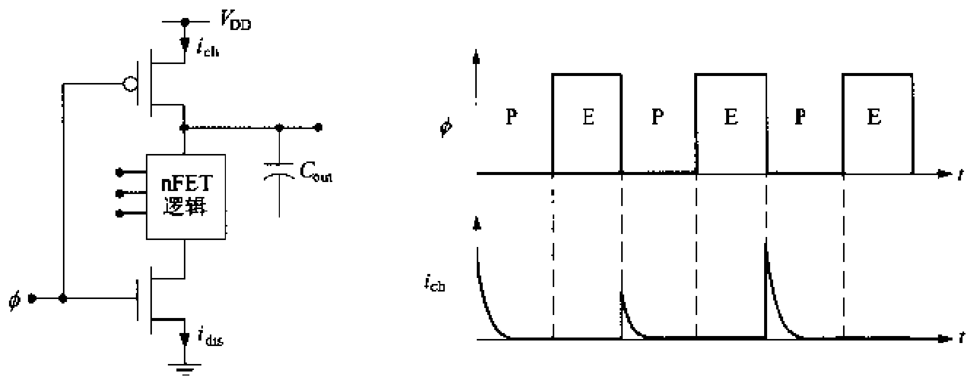


图 14.46 动态逻辑门中周期充电的电流

SSN 的大小也取决于连接芯片至引线的封装和压线技术。在多芯片模块(MCM)中,SSN 可以在不同的芯片间引起不希望的相互作用。这些及其他相关的问题是许多研究课题的焦点。

14.8 低功耗设计考虑

一个芯片的总功耗 P[W]在现代的 VLSI 中是极为重要的。如果采用的是电池供电,那么 P 决定了在需要重新充电之前可以工作的时间。甚至在桌上系统中,也必须使功耗较小以保证硅芯片不会熔化(最坏情况)以及系统的冷却方式是足够的。低功耗设计技术已在电路和系统级开发。相当多的研究用来分析和解决这些问题。

在数字 CMOS 电路中功耗有三个主要来源。

- DC 功耗 P_{DC},它是当输入稳定时从 V_{DD}至地的直接导通路径引起的。在标准静态 CMOS 逻辑电路中,漏电电流是这引起一部分功耗的原因。
- 开关功耗 P_{sw}是当输入改变时引起电源通过管子至地有直接的电流路径。它发生在电压传输曲线(VTC)的过渡部分并且曾在前面一节关于 SSN 问题的来源中讨论过。
- 动态翻转功耗 P_{dyn}是由于对电容节点的充电和放电。它可以由以下的一般公式进行估计:

$$P_{dyn} = \alpha C V^2 f \tag{14.86}$$

式中 C 是电容(法拉),α 是翻转率系数,V 是电压摆幅,而 f 是信号频率。
瞬态总功耗为以下的和:

$$P = P_{DC} + P_{sw} + P_{dyn} \tag{14.87}$$

式中每项的值随电路设计技术而有所不同,一些功耗在芯片的某些部分可以占到主要地位。

首先考虑 DC 漏电项。它可以写成如下简单形式

$$P_{DC} = I_{DDQ} V_{DD} \tag{14.88}$$

式中 I_{DDQ}是当输入不变时流过的静态漏电电流。一个管于的漏电电流值是与工艺有关的。芯片的总 I_{DDQ}随着管子的数目而增加,并且也取决于电路设计技术。静态 CMOS 逻辑门显示最小的静态漏电电流,对于芯片其值通常小于 10 μA 左右。所引起的功耗在几十微瓦(μW)的数量级上。尽管它在某些设计中可能较大,但它通常是这三种功耗中最小的。

开关功耗 P_{sw}是门输入信号发生转换的结果,它引起了从 V_{DD}至地的直接电流通路并且是 SSN 的来源。它发生在输出电压经历电压过渡的时候,并且也与电路设计有关。静态逻辑门总有开关功耗,因为不可能消除导电通路。一个简单的估计是:

$$P_{sw} = \langle I_{sw} \rangle V_{DD} \tag{14.89}$$

式中$\langle I_{sw} \rangle$是平均的 DC 电流。单独一个门的这个功耗随管子的宽长比而变化。因为(W/L)决定了通过 FET 的电流大小。实际的数值取决于输入波形的形状,这使它很难用收敛形式的方程进行计算。电路模拟则是最精确的。

动态功耗通常认为是最难处理的。一般的表达式

$$P_{dyn} = \alpha C V^2 f \tag{14.90}$$

表明 P_{dyn}与信号开关频率 f 成正比,所以它是随电路的速度而增加的。降低这一项数值的一种办法是降低电源电压 V_{DD},因为它是上式中 V 的最大(DC)值。这同时也减少了其他功耗的值。处理器核的电压已在 2 V 以下,而且还在继续推向更低的操作电压。降低电源电压对于电池操作的单元也是一个优点。

尽管这看上去好像是一个简单的技术,但它却在电路级上引起问题,即它会引起较慢的开关速度。自然这首先就抵消了降低 V_{DD}的目的。为了理解这一点,回想一下一个不饱和的 FET 其电流为:

$$I_D = \frac{\beta}{2}[2(V_{GS} - V_T)V_{DS} - V_{DS}^2] \tag{14.91}$$

因为在电路中的最高电压是 V_{DD},降低它就意味着 I_D 也会减少。I_D 的减少意味着它需要较长时间充电输出电容,从而同时增加上升和下降时间。这将降低门的开关速度。为了补偿这一影响,可以提高器件的互导:

$$\beta = \mu_n C_{ox}\left(\frac{W}{L}\right) \tag{14.92}$$

因为

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{14.93}$$

使栅氧厚度 t_{ox}减小将增加 β,所以改进工艺可以有助于这一点。否则必须增大沟宽 W 以保持速度。

文献中发表了许多新的和独特的方法来降低 VLSI 芯片中的功耗。这个问题本身通常是在电路设计层次和体系结构层次这两方面来解决。有兴趣的读者可以参照这些文献。在本书参考资料中列出了有关这个内容的几本书。

14.9 参考资料

[1] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[2] Anantha P. Chandrakasan and Robert W. Brodersen, **Low Power Digital CMOS Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[3] Dan Clein, **CMOS IC Layout**, Newnes, Woburn, MA, 2000.

[4] Sabih H. Gerez, **Algorithms for VLSI Design Automation**, John Wiley & Sons, Chichester, England, 1999.

[5] Bryan Preas and Michael Lorenzetti (eds.), **Physical Design Automation of VLSI Systems**, Benjamin-Cummings Publishing Company, Menlo Park, CA, 1988.

[6] Jan M. Rabaey, **Digital Integrated Circuits**, Prentice Hall, Upper Saddle River, NJ, 1996.

[7] Jan M. Rabaey and Massoud Pedram, **Low Power Design Methodol-**

ogies, Kluwer Academic Publishers, Norwell, MA, 1996.

[8] Michael Reed and Ron Rohrer, **Applied Introductory Circuit Analysis**, Prentice Hall, Upper Saddle River, NJ, 1999.

[9] Kaushik Roy and Sharat C. Prasad, **Low-Power CMOS VLSI Circuit Design**, John Wiley & Sons, New York, 2000.

[10] T. Sakurai and K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances," *IEEE Trans. Electron Devices*, vol. ED-30, no. 2, pp. 183-185, Feb. 1983.

[11] M. Sarrafzadeh and C. K. Wong, **An Introduction to VLSI Physical Design**, McGraw-Hill, New York, 1996.

[12] Ramesh Senthinathan and John L. Prince, **Simultaneous Switching Noise of CMOS Devices and Systems**, Kluwer Academic Press, Norwell, MA, 1994.

[13] Naved Sherwani, **Algorithms for VLSI Physical Design Automation**, Kluwer Academic Publishers, Norwell, MA, 1993.

[14] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

[15] M. Michael Vai, **VLSI Design**, CRC Press, Boca Raton, FL, 2001.

[16] Gary K. Yeep, **Practical Lower Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1998.

14.10 习题

[**14.1**] 考虑一条互连线的几何形状如图 14.3 所示，其中 $T_{ox}=1.10\ \mu m$，$w=0.5\ \mu m$，$t=0.90\ \mu m$。

(a) 利用简单的平行板公式计算每单位长度的电容 c[pF/cm]，忽略边缘电容。

(b) 用包括边缘电容的经验公式求出预测的 c 值。

(c) 假设在(b)部分的结果是正确的，求出忽略边缘电容时引起的百分误差。

(d) 这条互连线的薄层电阻为 $R_S=0.08\ \Omega$，如果这条线为 100 μm 长，求 R_{line}和 C_{line}的值。

[**14.2**] 一条互连线的几何形状如图 14.3 所示，其中 $T_{ox}=0.90\ \mu m$，$w=0.35\ \mu m$，及 $t=1.10\ \mu m$。

(a) 用包括边缘电容的经验公式求出预测的 c 值。

(b) 该互连线的薄层电阻为 $R_S=0.04\ \Omega$，如果线长为 48 μm，求 R_{line}和 C_{line}的值。

(c) 对该互连线建立一个 $m=7$ 的 RC 梯形等效电路，然后用这个模型来决定时间常数。把它与运用简单公式 $\tau=R_{line}C_{line}$得到的时间常数进行比较。

[**14.3**] 考虑一个 CMOS 工艺，它具有 4 层金属，其厚度如图 P14.1 所示。对每层金属求出它的单位长度电容 c[pF/cm](金属至衬底电容)。假设每一氧化层淀积后进行了 CMP。

[**14.4**] 计算图 P14.1 所示的 M1 – M2 层每 μm^2 的重叠电容。对 M3 – M1 层重复这一计算。

[**14.5**] 一条互连线的几何形状如图 14.3 所示。重要的参数值为 $T_{ox}=1000$ nm，$w=0.35\ \mu m$，$t=0.85\ \mu m$，$R_S=0.008\ \Omega$ 及 $l=122\ \mu m$。

(a) 求 R_{line}和 C_{line}的值。在你的计算中都要包括边缘效应。

(b) 建立 $m=2$ 段，$m=6$ 段的 RC 梯形等效电路。求出这两种等效电路的时间常数差别。然后将 $m=6$ 段的结果与方程(14.19)进行比较。

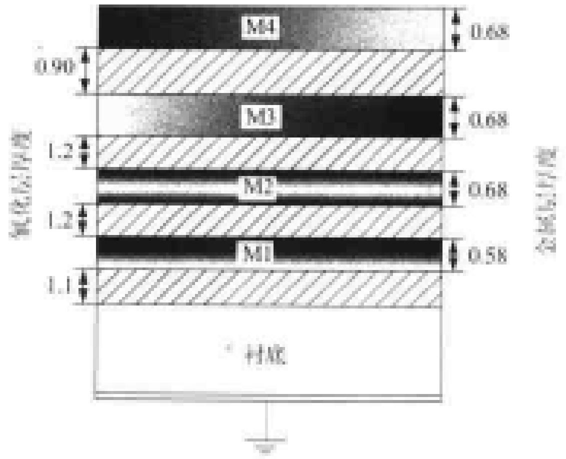


图 P14.1

[**14.6**] 一条互连线描述为 $w=0.35\ \mu m$，$T_{ox}=1.20\ \mu m$，$t=0.95\ \mu m$，及 $R_s=0.008\ \Omega$

(a) 计算 r 的值[Ω/cm]和 c 的值[pF/cm]。

(b) 假设在方程(14.33)中 $\xi=0.05$，求出表示互连线时延的方程。

(c) 计算导线长度为 100 μm，200 μm 和 300 μm 时的延时(ps)。

[**14.7**] 在一个互连层上的两条线相隔的间距为 $S=0.50\ \mu m$。每条线的 $w=0.35\ \mu m$，$T_{ox}=1.1\ \mu m$ 及 $t=1\ \mu m$

(a) 计算每单位长度的耦合电容 c_c。

(b) 如果相互作用的长度为 20 μm，求耦合电容。假设相互作用的长度为 30 μm，重复这个问题。

[**14.8**] 考虑两条互连线相隔 $S=0.40\ \mu m$。每条线的 $w=0.25\ \mu m$，$T_{ox}=1.2\ \mu m$ 及 $t=0.85\ \mu m$

(a) 计算一条线每单位长度的自身电容 c。

(b) 计算在这两条线之间每单位长度的耦合电容 c_c。

(c) 假设两条线均为 18 μm 长，求出从这两条线中的一条看进去的总电容。

[**14.9**] 一条互连线描述为 $w=0.4\ \mu m$，$T_{ox}=1.0\ \mu m$，$t=0.84\ \mu m$，及 $R_S=0.005\ \Omega$，它的长度为 50 μm

(a) 求出这条线每单位长度的电容、总电容和线的电阻。

(b) 假设在方程(14.50)中那样宽度和长度都按 $s=1.5$ 的因子缩小。材料和氧化层的厚度值保持不变。对这条尺寸缩小后的连线求出新的线电容和电阻值。

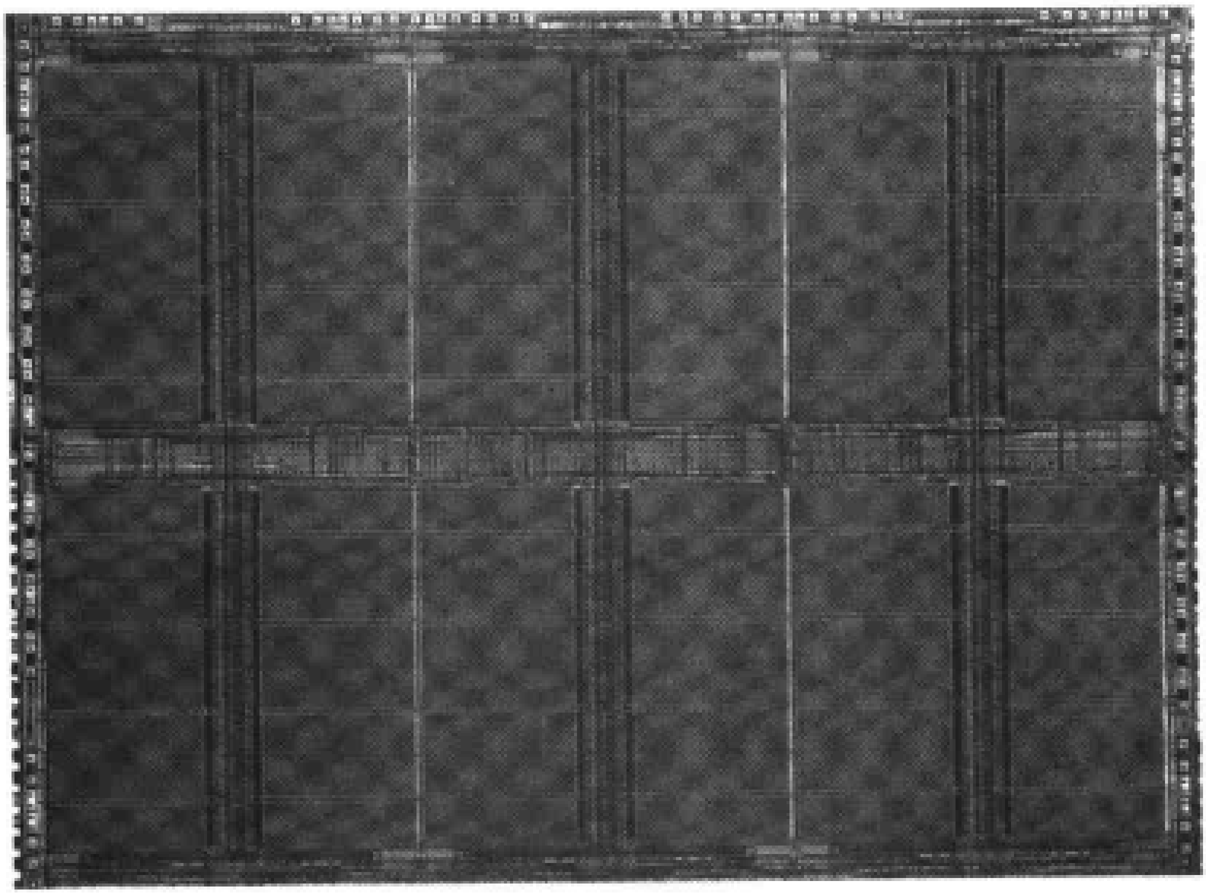
[**14.10**] 考虑描述在分布 RC 线上 $v(z,t)$的方程(14.30)。在 erfc (ξ)函数中的 ξ 值选为 0.9。

(a) 用电源电压 V_{DD}来表示的 $v(z,t)$值是什么？[erfc (ξ)的数值可以在数学用表或在大多数 PC 计算程序中找到]。

(b) 这条线的参数是 $w = 0.5\ \mu m$, $T_{ox} = 0.9\ \mu m$, $t = 0.90\ \mu m$, 以及 $R_s = 0.04\ \Omega$。对于所选的 ξ 值求出在延时方程 $\tau = Bl^2$ 中的 B 值。

[**14.11**] 图 14.33 所示的施密特触发器电路采用 $V_{DD} = 3.3$ V 的电源。晶体管的阈值电压为 $V_{Tn} = 0.7$ V 及 $|V_{Tp}| = 0.8$ V。如果器件比为 $(\beta_1/\beta_3) = 6$ 及 $(\beta_4/\beta_6) = 4$, 计算 V^+ 和 V^-。

[**14.12**] 考虑一个图 14.33 所示的施密特触发器电路, 它的制造工艺为 $V_{DD} = 5$ V, $V_{Tn} = 0.7$ V 及 $|V_{Tp}| = 0.8$ V。设计这个电路使 $V^+ = 3.9$ V 及 $V^- = 1.2$ V。



第 15 章　VLSI 时钟和系统设计

同步设计采用时钟信号,使数据通过系统的移动保持一致。VLSI 电路的整体速度常常是由时钟确定的限度来决定的。本章中将考察 CMOS 情况下的同步逻辑设计。

15.1　时钟控制触发器

最简单的 CMOS 时钟技术基于单个时钟信号 $\phi(t)$,其周期为 T 秒而频率 $f=(1/T)$ 赫兹。一个理想的时钟波形如图 15.1 所示。其幅度从 0 变化到 1,相应的电压范围为 $[0, V_{DD}]$。在经典的数字设计中,数据流是通过时钟控制锁存器或触发器的装载而达到同步的。图 15.2 描述了 D 型触发器的这一情形。数据位 D 只是在时钟上升沿时装入 DFF 中。如果上升沿发生在 t_0 时刻,那么经过一段时延 t_{ff} 之后输出端就会出现这一值:

$$Q(t_0+t_{ff}) = D(t_0) \tag{15.1}$$

数据的获得会在下一个上升沿(图中为 t_1)以及之后在每个 0 至 1 的时钟过渡时再次发生。限制电路的因素是 DFF 的延迟时间 t_{ff},它是由电子电路及负载情况决定的。减少 t_{ff} 允许较高频率的时钟,从而提高数据流通率。同样的情况也适用于负边沿触发的 DFF。其他类型的触发器,如 JK 触发器,也可以用 CMOS 构成,但却很少用在高密度的设计中,因为它们较慢而且占用较多的面积。

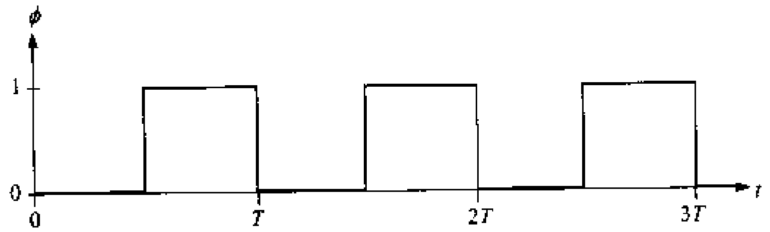


图 15.1　理想的时钟信号

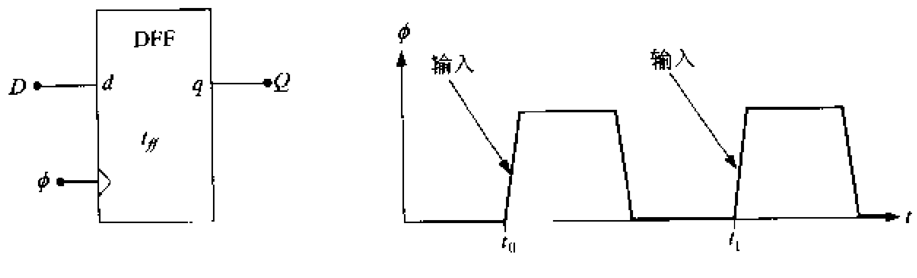


图 15.2　DFF 的时序

经典的状态机

时钟控制的触发器提供了经典的时序逻辑电路的基础。图 15.3 为采用单时钟时序状态机的两个模型。图 15.3(a)的 Moore 机器把输入送到组合逻辑构成的输入逻辑块，其输出送到一个寄存器。寄存器本身为 m 位宽并且能用触发器构成；这使该状态机具有 2^m 个状态。状态寄存器的输出送到输出逻辑块中以产生输出数据(Output)。它们也被送回到输入逻辑块。由于被存放的数据延迟了一个时钟周期，所以该状态机的当前状态会影响下一个状态。图 15.3(b)中的 Mealy 机具有相同的基本结构，但允许当前的输入直接影响输出。

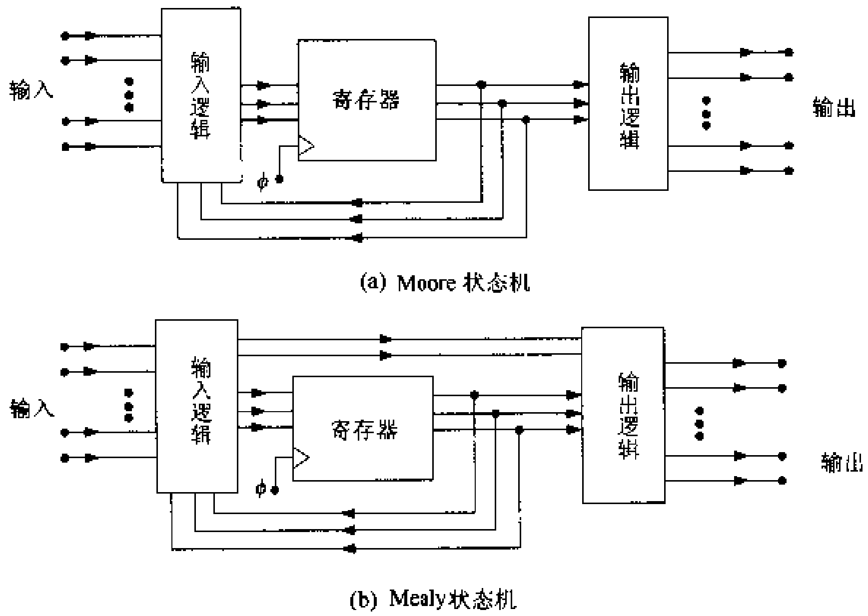


图 15.3　Moore 和 Mealy 状态机

图 15.4 所示的 Huffman 模型中的一般化结构同时包含 Moore 和 Mealy 模型且把它们作为特殊情况。外加的原始输入 $x_0, x_1, \cdots, x_n$ 和从存储单元来的二次输入 $q_0, \cdots, q_k$ 都送到组合逻辑块中。原始输出 $f_0, f_1, \cdots, f_m$ 来自于组合逻辑块。二次输出 $d_0, d_1, \cdots, d_k$ 用来作为钟控 DFF 的输入；它们在下一时钟周期时作为二次输入。

采用任何这些经典模型可以设计并实现 VLSI 时序逻辑电路。这是在逻辑级设计基于单元的 ASIC 的一个最常用方法，即工程师从状态图和逻辑电路开始设计。CAD 工具用来综合电路，而布局和布线则进行平面布局和互连线的布线。硅片上的细节对于设计者来说是看不见的。确实，先进的工具包允许我们设计 ASIC 而不需要了解任何有关物理设计的知识。这使它们在快速完成原型设计方面非常有用，并且只要速度可以接受，它们也可以用于大批量生产。

FPGA 设计很大程度上也基于经典的状态机理论，但通常要求更深刻理解 VLSI 的概念。组合逻辑电路设计的电路类型随商家而不同。普遍采用的有各种门，PLA，可编程逻辑器件(PLD)及多路选择器组等。编程是通过 EPROM，熔丝和 SRAM 阵列来完成的，有些还包含查表(LUT)帮助设计。虽然芯片级的设计是自动完成的，但也有可能由设计者手工完成分割，单元布置和布线以提高电路的速度。无论 ASIC 还是 FPGA 的设计规格都取决于商家，所以有兴趣的读者可以去看具体器件的文献了解细节。这里将介绍有关设计问题的一般观点，以

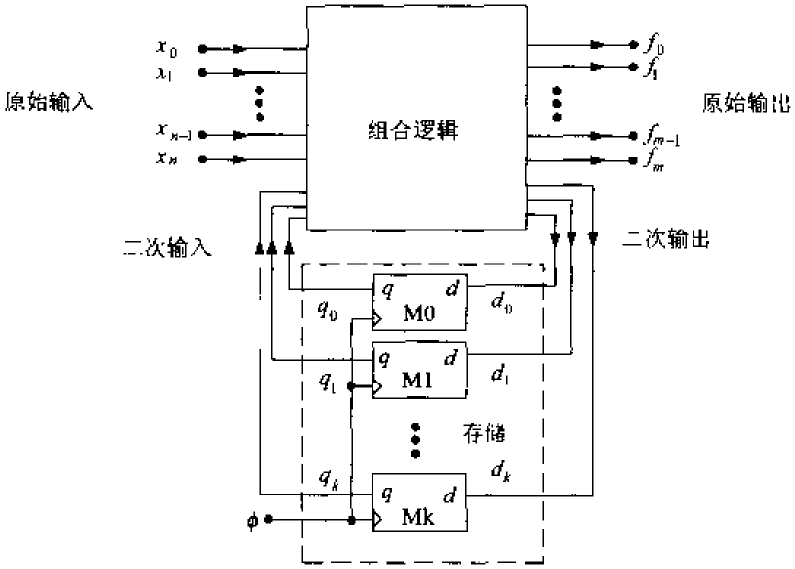


图 15.4 Huffman 状态机模型

说明电路和物理设计层的重要特点。

经典的状态机用在各种类型较大的 VLSI 芯片中。设计方法本身虽然依据标准的流程，但通常都较紧密地联系着物理设计。在基于单元的 VLSI 芯片中，触发器作为预先定义的模块从库中例举；逻辑电路可以用基本的门或逻辑阵列来设计。物理设计的关键部分集中在门和互连线的延迟以及时钟。

让我们以 Huffman 模型为基础，研究一个状态机的结构以说明主要的概念。图 15.5 为采用 AND-OR 可编程逻辑阵列实现组合逻辑的方块图。PLA 的 AND 平面可以通过编程用输入 x_0，…，x_n和 q_0，…，q_k产生积项(小项)。而 OR 平面则给出了如下的 SOP(积项和)的输出：

$$\sum_r m_r \tag{15.2}$$

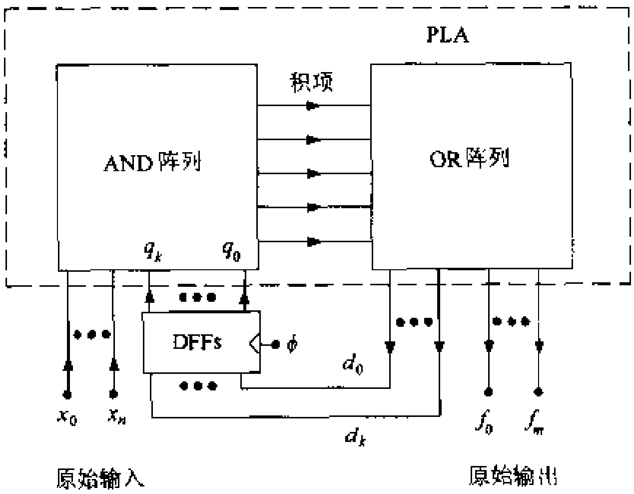


图 15.5 采用 PLA 逻辑实现的 Huffman 状态机

其中有些输出为原始输出 f_0，…，f_m，而其余则作为二次输出 d_0，…，d_k反馈且存放在 DFF 中，在下一个时钟周期时作为二次输入。逻辑设计的细节与在标准数字系统中遇到的完全一

样;VLSI 方面的考虑则出现在 PLA 中,时钟周期 T 必须足够大以允许完成逻辑周期。这一条件可以表示为:

$$T > t_{ff} + t_d + t_{su} \tag{15.3}$$

式中

- t_{ff}是从触发器的输入至输出的延迟时间,
- t_d是通过 PLA 的逻辑延迟时间,
- t_{su}是触发器的"建立时间",也就是在输入被锁存入 DFF 之前它必须稳定的时间。

因为逻辑阵列也许要求较长的互连线,因此电路有可能不够快而不能满足所希望的速度。随机组合逻辑通常可以通过减小 t_d而实现较快的电路,但这需要较长的设计时间。另一种方法是改变触发器以减小 t_{ff}和t_d的值。

这一简单的电路例子说明了 VLSI 可以很容易地适合基于触发器存储的经典状态机的结构。然而 CMOS 电路可以提供比其他工艺可能提供的更为复杂的时钟和时序策略。当更为详细地研究时,在 VLSI 中同步逻辑的全部功能就会变得十分明显。

15.2 CMOS 时钟方式

CMOS 允许采用范围很广的各种时钟方式。有些是非常一般的,而另一些则基于电路设计技术。本节中我们将考察在 VLSI 环境中用来同步数据流的一些最为常用的方法。

15.2.1 钟控逻辑链

最简单的是基于采用单个时钟 ϕ,或者就是这个时钟本身(单相位设计),或者连同它的反信号 ϕ(两相位系统),如在第 2 章中所介绍的那样。图 15.6 为这一对时钟信号的理想波形。用电压来表示时,我们有:

$$V_{\phi} = V_{DD} - V_{\phi} \tag{15.4}$$

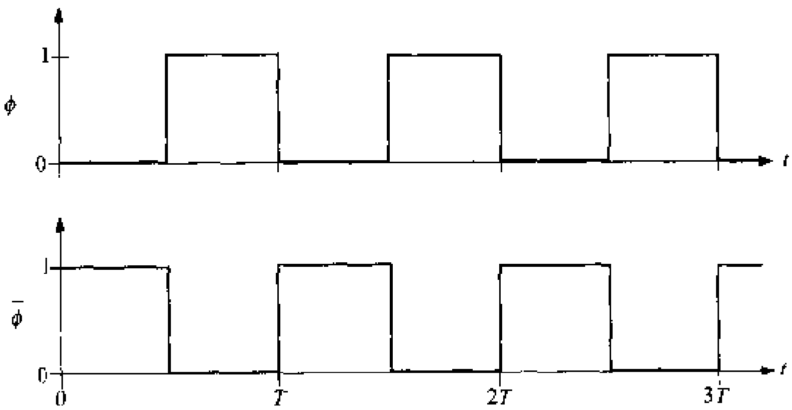


图 15.6　互补时钟

因为这两个信号都假设是全轨信号,它们的范围从 0 V 到 V_{DD}。一对不重叠的时钟信号在所有的时刻 t 总应满足

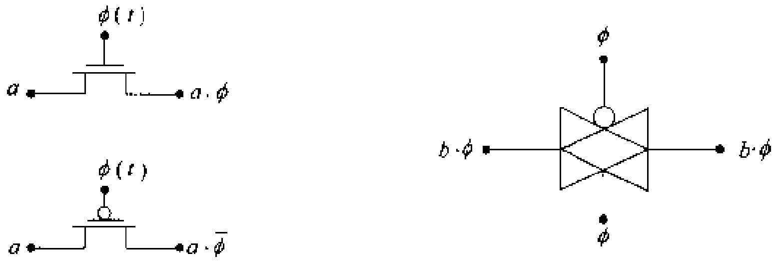
$$\phi \cdot \overline{\phi} = 0 \tag{15.5}$$

但这对于互补信号并不完全正确，这是由于在上升/下降期间必然存在重叠所致。

把时钟信号直接加到 FET 上是控制数据流的一种简单方法。图 15.7 为三种主要的时钟器件。可以采用图 15.7(a)中单极性管子开关中的任一种，但一般选择 nFET 而不是 pFET，这是由于 nFET 有较好的导电特性。采用 nFET 作为时钟器件的一个问题是输出被限制在 $[0, V_{max}]$ 的电压范围内，这里

$$V_{max} = V_{DD} - V_{Tn} \tag{15.6}$$

这是由于阈值损失所致。除了输出一个弱的逻辑 1 以外，过渡也很慢。图 15.7(b)的传输门无论对于输入逻辑 0 还是逻辑 1 均能较快地切换并通过全电压范围 $[0, V_{DD}]$。然而一个 TG 需要两个管子(一个 nFET 和一个 pFET)以及与此相关的并行连线增加了布线问题。在现代高速设计中，往往采用单个的 nFET 而不是 TG，这是因为单个的 nFET 比较简单；而传输逻辑 1 的问题则通过仔细设计外围电路来解决。因为钟控 TG 电路已在第 2 章中学习过，在这里将集中在钟控 nFET 上。如果希望，每一个电路都可以重新设计成包含 TG 在内。



(a) FETs (b) 传输门

图 15.7 时钟控制晶体管

考虑图 15.8 的逻辑链。数据在组合逻辑(C/L)单元之间的传送通过时钟控制的 nFET，其中 ϕ 控制输入进入单元 1 和 3，而 $\overline{\phi}$ 控制数据进入单元 2。从时钟 $\phi = 1(\overline{\phi} = 0)$ 开始可以看到同步的数据流。输入 $a_0, \cdots, a_3$ 被允许进入单元 1，且在经过一段特征延时之后将出现在这个逻辑块的输出端上。当时钟变为 $\phi = 0(\overline{\phi} = 1)$ 时，$b_0, \cdots, b_3$ 传送到单元 2 中。在下半个时钟周期，ϕ 上升至 1(所以 $\overline{\phi} = 0$)，于是允许 $c_0, \cdots, c_3$ 进入单元 3。这一链的输出 $d_0, \cdots, d_3$ 在经过一段逻辑延时后就可以得到。这一时钟方式在每半个时钟周期内把数据从一个单元送到下一个单元，因此数据流非常直观明显。

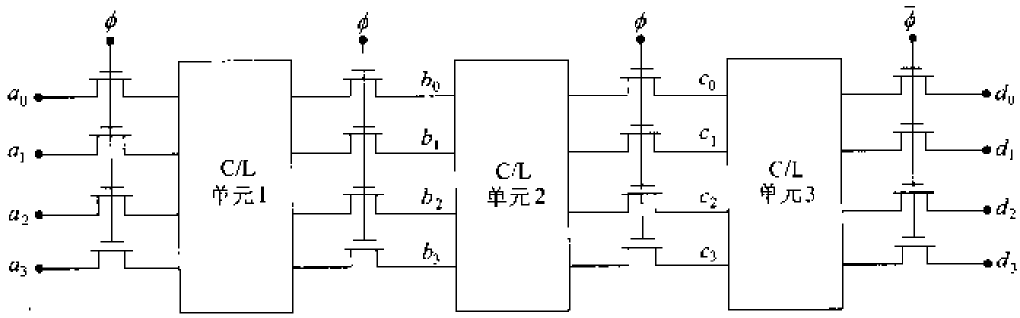


图 15.8 时钟控制的逻辑链

1. **时序圆图和时钟偏差(Timing Circles and Clock Skew)**

时序圆图是一个简单的结构,它在观察数据传输时非常有用。图 15.9 显示了**钟控逻辑链**的时序圆图。两个时钟信号 ϕ 和 $\bar{\phi}$ 的周期为 T,并且都假设为在半个时钟周期($T/2$)内具有逻辑值 1,另半个时钟周期内具有逻辑值 0。这定义了所谓的 50%工作周期(占空比)。由于时钟每个周期重复,我们建立一个时序圆图,它的一个整圈代表 T,然后在这个图的各部分标注上相应周期时钟逻辑值为 1 的时钟信号。图中显示了一对时钟信号($\phi,\bar{\phi}$)的时序圆图。引入一些术语来描述圆图中所包含的信息是很有用的。"在 ϕ 期间"这一说法是指 $\phi=1$ 的时间。类似地,把"在 $\bar{\phi}$ 期间"理解为 $\bar{\phi}=1$ 的时间间隔。当应用到图 15.8 的钟控逻辑链中时,时序圆图就显示进入每类单元中的数据流。在 ϕ 期间,输入为奇数单元所接受,它们在图中被称为"ϕ 输入部分"。类似地,偶数单元 2(即"$\bar{\phi}$ 输入部分")在 $\bar{\phi}$ 期间接受输入。在逻辑链中 ϕ 和 $\bar{\phi}$ 部分相互交替形成了数据同步移动通过系统。

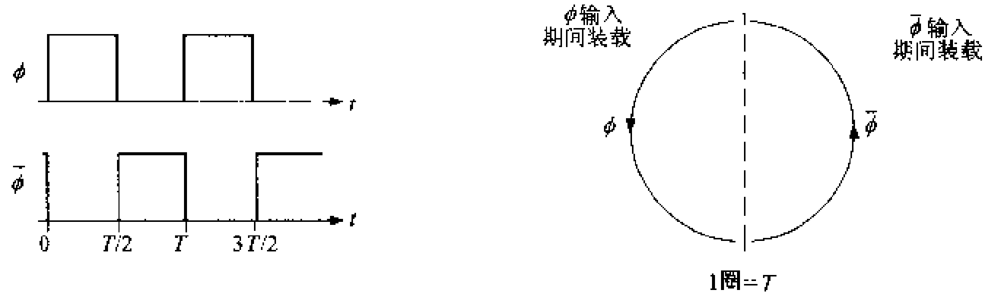


图 15.9　单时钟两相位逻辑链的时序圆图

钟控逻辑链的简单性使它作为一种基本设计方法学十分吸引人。然而当转入电路和物理设计层时,必须考虑几个复杂的因素。影响所有钟控系统的一个关键问题是**时钟偏差**。时钟偏差是时钟的时序与系统的参照在相位上不一致。它可以来源于不同的原因,并且限制时钟的频率。在同步系统中,这相当于限制了数据流通率及整个系统的速度。

让我们来考察一下在时钟产生电路中出现的时钟偏差。图 15.10 的电路采用反相器从时钟信号 Clk 中产生 ϕ 和 $\bar{\phi}$。如果连线电容 C_1和 C_2相等,那么 ϕ 会比 $\bar{\phi}$ 稍有延迟,相差一个反相器的延时 t_d。这就确定了在图 15.11 的时钟波形中所显示的偏差时间 $t_s=t_d$。注意由于时钟偏差会增加重叠。这在数据同步和数据流中会引起问题,所以已经做出许多努力来使 t_s相对于时钟周期 T 的值达到最小。电路对于时钟偏差的敏感程度取决于设计风格。

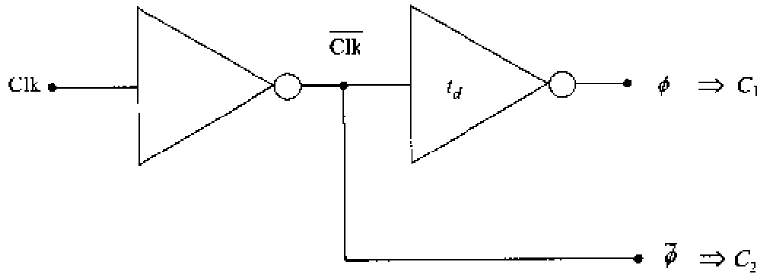


图 15.10　时钟产生电路

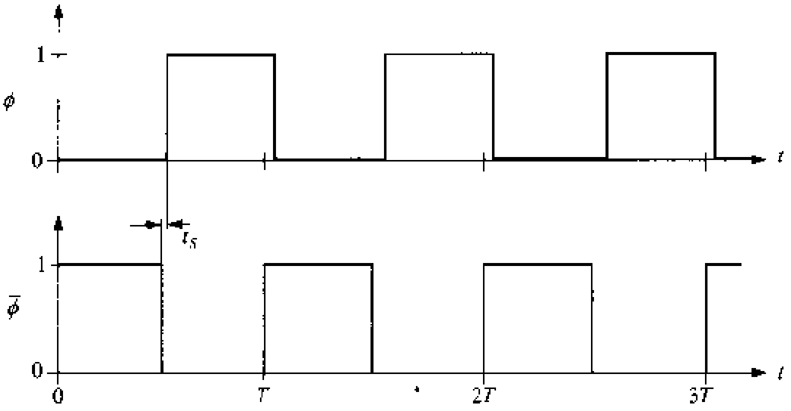


图 15.11 时钟偏差

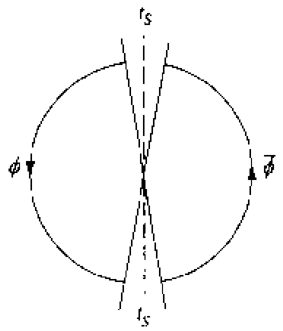


图 15.12 考虑时钟偏差时的时序圆图

通过把时序圆图修改成如图 15.12 所示的那样，可以看到时钟偏差的全面影响。在这个方法中，时钟偏差在画成虚线的垂直参考轴左右均匀分布。偏差时间 t_s减少了 ϕ 和 $\bar{\phi}$ 传送数据的时间。这也许要求采用一个较慢的时钟频率以允许逻辑单元处理数据。通过设计时钟分配网络，在时钟产生电路中引起的时钟偏差可以被控制到有限的程度。这相当于改变图 15.10 中所示的 C_1和 C_2值，这内容将在 15.4 节中讨论时钟分配问题时较为详细节地介绍。

2. 电路影响与时钟频率

钟控逻辑链的逻辑级描述没有显示出决定最高速度的电路特性。因为通过逻辑链的数据传送速率是由时钟频率决定的，所以理解限制速度的电子电路方面的原因是很重要的。这可以用图 15.13 所示的移位寄存器来说明。这一操作是很容易理解的。时钟 $\phi=1$ 的条件允许输入 a 进入第一级。在本例中"逻辑单元"就是一个反相器。当工作周期(占空比)为 50%时，$(T/2)$的时间被分配来传送数据。在这个时间内必须有两件事发生。首先，a 的电压值必须通过 nFET 到达反相器的输入端。第二，反相器必须对这个输入做出响应并且产生输出 $\bar{a}$，这可写出如下的条件：

$$\left(\frac{T}{2}\right)_{\min} = t_{\mathrm{FET}} + t_{\mathrm{NOT}} \tag{15.7}$$

上式即为允许最小的半周期时间。在这个方程中，t_{FET}是通过传输管的延时而 t_{NOT}是门延时。由于通过一个 nFET 的最坏情形是传送逻辑 1，我们有：

$$\left(\frac{T}{2}\right)_{\min} = t_{\mathrm{r,FET}} + t_{\mathrm{HL,NOT}} \tag{15.8}$$

式中 $t_{\mathrm{r,FET}}=18\tau_{\mathrm{FET}}$是 nFET 的延时而 $t_{\mathrm{HL,NOT}}$是 NOT 门的从高至低的时间。因此该移位寄存器的最高时钟频率为：

$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{2(t_{\mathrm{r,FET}} + t_{\mathrm{HL,NOT}})} \tag{15.9}$$

移位寄存器的结果可以延伸到随意的逻辑链，即可以写出：

$$f_{max} = \frac{1}{T_{min}} = \frac{1}{2(t_{r,FET} + t_{CL})} \qquad (15.10)$$

式中 t_{CL}代表在逻辑链中最长的组合逻辑延时。这清楚地说明了系统的时钟速度是如何由门的延时决定的。最大的性能要求最坏情况的逻辑路径要尽可能地快。对于一组设计，这是在电路层次上通过仔细选择电路设计类型或版图来达到的。另一个方法也许值得把逻辑块分成几个较小的部分，然后重新设计逻辑链。

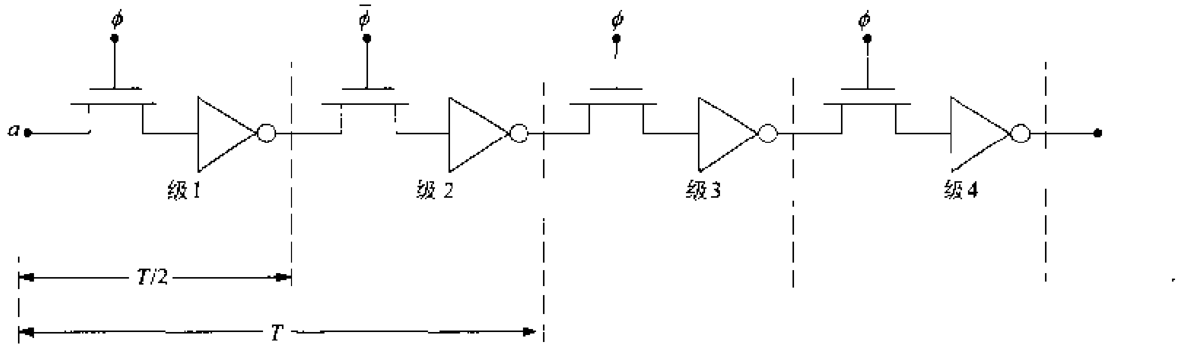


图 15.13 移位寄存器电路

尽管不明显，但这个 CMOS 设计方式在本质上是动态的并且显示出漏电问题。考虑一个逻辑 1 的电压传送通过 nFET 然后时钟变为 $\phi = 0$ 的情形。这显示在图 15.14(a)中。尽管 nFET 截止，但仍有漏电电流 I_{leak}，它从电容 C_{in}中移去电荷，如下式所描述[①]

$$I_{leak} = -C_{in}\frac{dV_{in}}{dt} \qquad (15.11)$$

漏电电流是电压 V_{in}的函数，所以上式是一个非线性的微分方程。假设初始条件 $V_{in}(0) = V_{max}$，于是就得到了类似于图 15.14(b)所示的电压衰减情形。图中 V_1表示反相器识别输入为逻辑值 1 所需要的最小电压。维持时间 t_h是保持这个输入状态的限制因素。当时钟的工作周期为 50% 时，这意味着：

$$\left(\frac{T}{2}\right)_{max} = t_h \qquad (15.12)$$

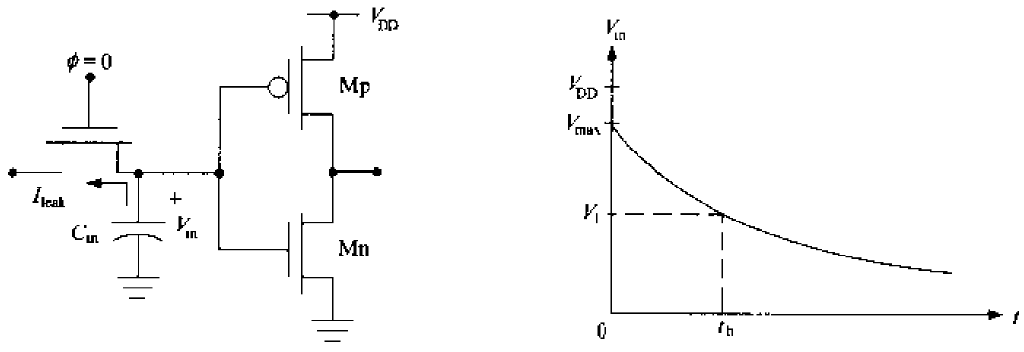
这是因为时钟有半个周期为 0。这就确定了最低的时钟频率 f_{min}为

$$f_{min} = \frac{1}{T_{max}} = \frac{1}{2t_h} \qquad (15.13)$$

如果 $f < f_{min}$，那么数据就会被破坏。虽然这个考虑并不影响高速电路，但它确实说明了存在对最低时钟频率的要求。这一特性的一个结果是时钟不能像在静态电路那样在测试时停止。

维持时间取决于漏电电流、输入电容以及 V_1的值。减小电荷漏电影响同时也增加 f_{max}的一种电路技术是把反相器设计成具有较小中点电压 V_M的值。因为 V_M是在 0 和 1 电压范围之间，减小它也就减小了 V_1。为了设计这一电路，回想一下：

① 电荷泄漏已在第 9 章 9.5 节中讨论。



(a) 电路　　(b) 电压衰减

图 15.14　移位寄存器中的电荷泄漏

$$V_M = \frac{V_{DD} - |V_{Tp}| + \sqrt{\frac{\beta_n}{\beta_p}} V_{Tn}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{15.14}$$

于是就得到用晶体管互导比表示的反相器的 V_M:

$$\frac{\beta_n}{\beta_p} = \frac{k'_n\left(\frac{W}{L}\right)_n}{k'_p\left(\frac{W}{L}\right)_p} \tag{15.15}$$

如果采用尺寸相同的器件,上式可简化为:

$$\frac{\beta_n}{\beta_p} = \frac{k'_n}{k'_p} \tag{15.16}$$

因此 $V_M<(V_{DD}/2)$。这也可以应用到任何静态逻辑门中,但必须小心以保证门的开关速度并不增加许多。

应当记住,时钟脉冲在实际中并不是阶跃而是具有一定的上升和下降时间如图 15.15 所示。在重叠期间,ϕ 和 $\bar{\phi}$ 控制的 FET 将会同时部分导通。这会引起**信号竞争**问题,即下一次的输入值竞先通过一个组合逻辑块并且产生一个不正确的输出传送到下一级。在设计的模拟

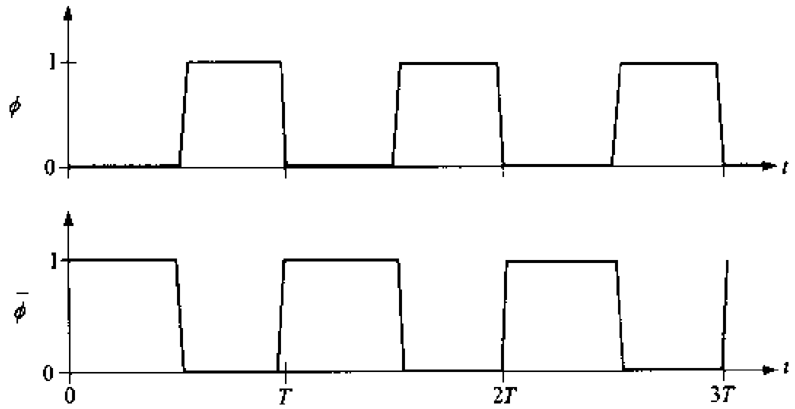


图 15.15　具有一定上升和下降时间的时钟波形

和验证阶段,必须检查这些情形。

图 15.16 的移位寄存器通过提供静态反馈回路避免了电荷泄漏的问题。这个设计的缺点是增加了逻辑门的数目以及反馈回路的布线。对这个电路的简单观察可知,它就是一个修改的主从 DFF 电路。组合逻辑块可以放在图面上由虚线确定的各级之间以构成钟控逻辑链。

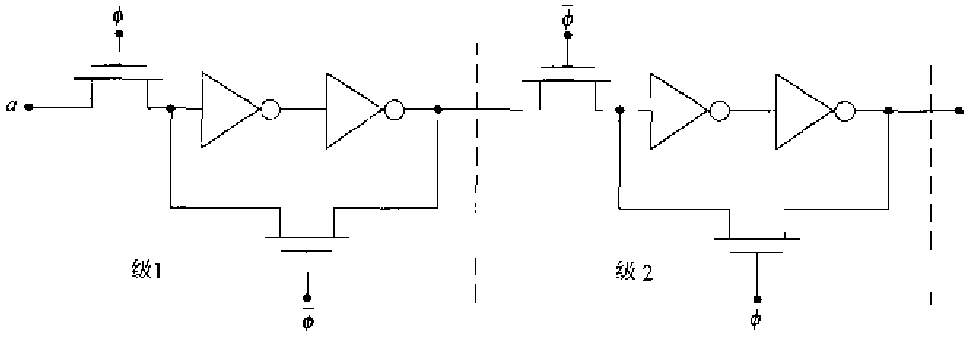


图 15.16 静态移位寄存器设计

3. 两相不重叠时钟

这个技术中采用两个不重叠的时钟 ϕ_1 和 ϕ_2,在所有的时刻 t 它们满足

$$\phi_1(t)\cdot\phi_2(t) = 0 \qquad (15.17)$$

它类似于单个相位时钟的方法但采用小于 50% 的工作周期值。图 15.17 显示了典型的一组波形。它可以用与 $\phi,\bar{\phi}$ 一对时钟相同的方式来控制通过一个逻辑链的数据流。两相不重叠时钟系统的时序圆图显示在图 15.18;由于脉冲宽度变窄,所以数据传送时间减少了。同时由于保持着不重叠的条件,所以信号竞争被消除了。

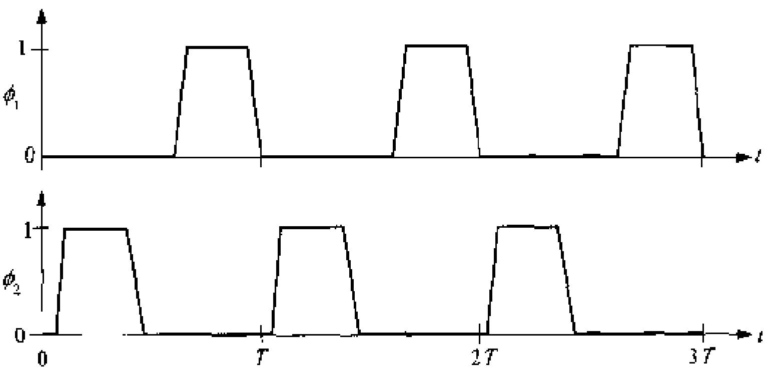


图 15.17 两相不重叠时钟

基于两相时钟技术的有限状态机可以提供功能很强的交互作用能力。比较简单的结构之一显示在图 15.19 中。它包括两个不同的逻辑块,逻辑块 A 和逻辑块 B 通过各自被控制的反馈寄存器连接起来。在这一结构中,逻辑块 A 的二次输出送入到 ϕ_1 控制的寄存器,后者提供了送入逻辑块 B 的二次输入。这一结构左边 ϕ_2 控制的寄存器从逻辑块 B 取得信号并把它们送入到逻辑块 A。其他不同的结构类型可以包括附加的反馈路径,把寄存器连到它自

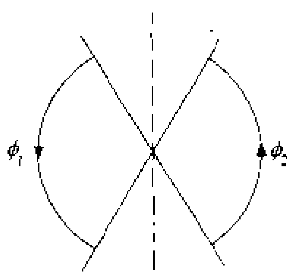


图 15.18 两相时钟系统的时序圆图

己的逻辑块，例如一组 ϕ_1（寄存器）的输出连回到逻辑块 A。采用这一方法的主要困难是产生这两个时钟本身，因为它们必须从单个的参考信号中推导出来。

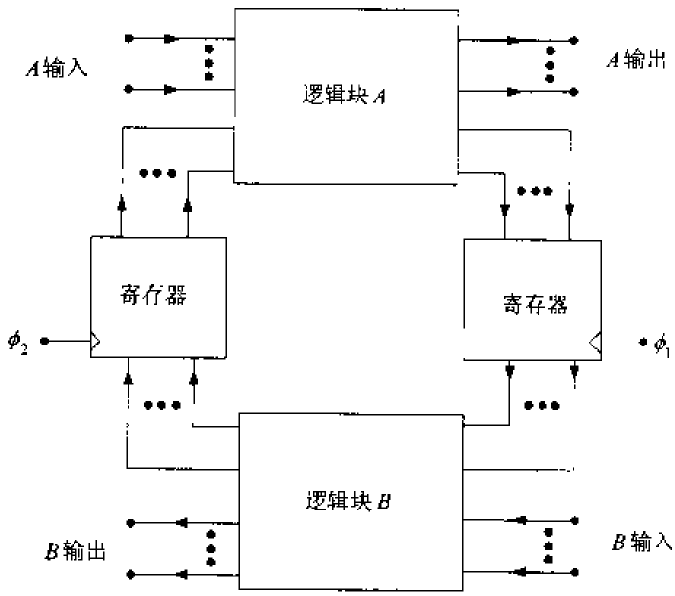


图 15.19 两相时钟有限状态机的设计

4. 其他多时钟技术

还有可能构成各种不同的多时钟技术，控制时钟控制逻辑链和状态机。例如，三个不重叠的时钟组合可以具有图 15.20 所示的波形。图 15.21 为它的时序圆图。以类似的方式也可以得到一个四时钟的组合。这些和其他时钟技术多年来已用在数字 MOS 集成电路的设计中。其中几个曾成功地用在基于较老的数字设计类型的商品芯片中，作为系统控制时钟。一些三相位和四相位的时钟技术也用于新的仅用 pMOS 和 nMOS 的低频动态逻辑电路中。

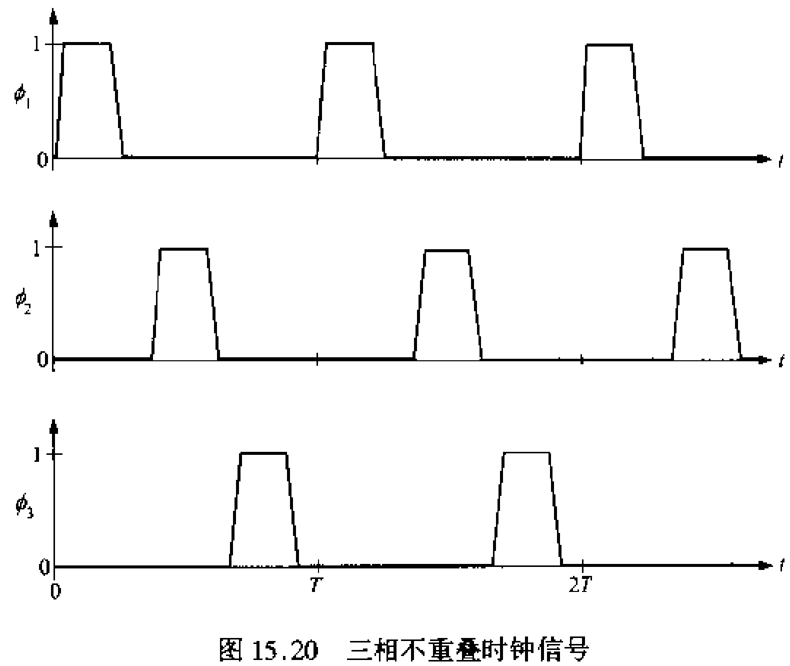


图 15.20 三相不重叠时钟信号

在现代高速 VLSI 中，复杂的时钟技术会带来许多问题，使它们不值得使用。速度的提高是通过改进电路设计，修改工艺和结构达到的。这对于系统时钟为 1 GHz 或更高的系统尤其如此，此时以较简单为好。在 VLSI 设计中最普遍的方法是运用单个时钟两相位系统。它能工作得很好并且允许改变体系结构而不必改变电路的设计类型。因此，将把研究局限在简单的时钟技术。

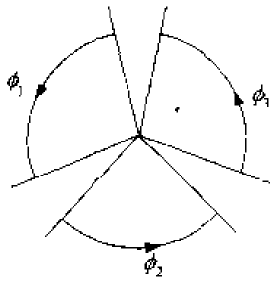


图 15.21 三相不重叠时钟系统的时序圆图

15.2.2 动态逻辑链[②]

动态逻辑电路通过控制逻辑门电路的内部工作状态达到同步的数据流。尽管动态逻辑链可以直接接口较简单的钟控逻辑电路，但它们的时钟策略是不同的。

让我们作为一个典型，回顾一下图 15.22 中多米诺逻辑级的操作。当时钟值为 $\phi=0$ 时，这一级被预充电(P)，此时 Mp 导通而 Mn 截止。内部节点电容 C 充电至 $V=V_{DD}$值，而该级的输出电压 $V_{out}=0$。求值发生在当时钟切换成 $\phi=1$ 时，pFET 此时被驱动至截止，而 Mn 则导通；nFET 逻辑阵列的输入在这期间有效。如果输入造成这个逻辑块为开路的开关状态，那么 V 就维持在高电平，该门的输出电压值和逻辑值均为 0。这个输出连接到下一级的一个逻辑 nFET，因此 $V_{out}=0$ V 将保持这个 nFET 截止。反过来，逻辑阵列从上到下的闭合开关状态使 C 放电，于是 $V=0$ V，输出因此切换为逻辑 1 的电压 $V_{out}=V_{DD}$。它把下一级的逻辑 FET 驱动至导通。以上可以用来说明时钟自动控制数据流的过程，因为输入和输出只是在求值期间才有效。

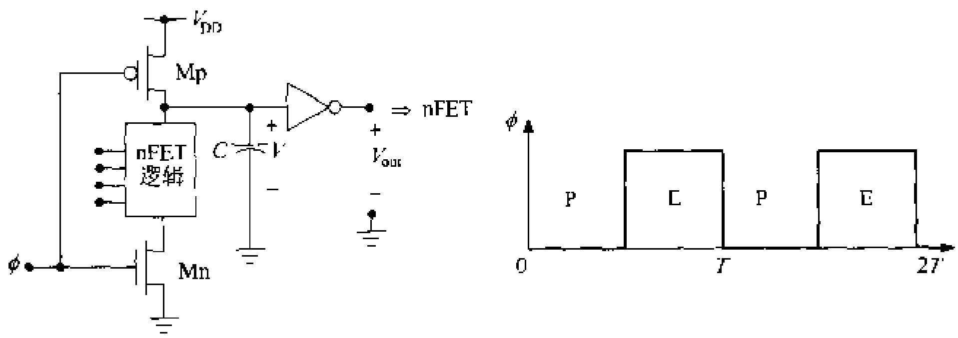


图 15.22 多米诺逻辑的工作情况

动态 CMOS 系统的时序可以通过对逻辑链进行相同的分析来理解。考虑图 15.23 显示的逻辑链，图中各级具有相同的基本多米诺结构。送入第 2 至第 4 级的每个输入都假设来自一个多米诺门，但在图中没有明示出来。单个时钟波形 ϕ 加到这个链中的每一级，所以这个链的行为如同单个逻辑组合。图 15.24 的波形同时显示这个链预充电和求值其间的行为。当 $\phi=0$ 时，每级同时预充电，不发生任何数据传送。求值发生在 $\phi=1$ 时，此时级 1a 和级 1b 的输入都假设为有效，所产生的输出连同从其他门来的输入一起送到级 2a。级 2a 所产生的结果连同级 2b 的输出一起传送到级 3a。这一连串的作用通过其余各级继续下去，直到最终结果 f 有效时为止。在时序图中，这一连串作用表示为时钟波形的求值期间被分成了几部分。

② 本节整个基于第 9 章 9.5 节所介绍的内容。

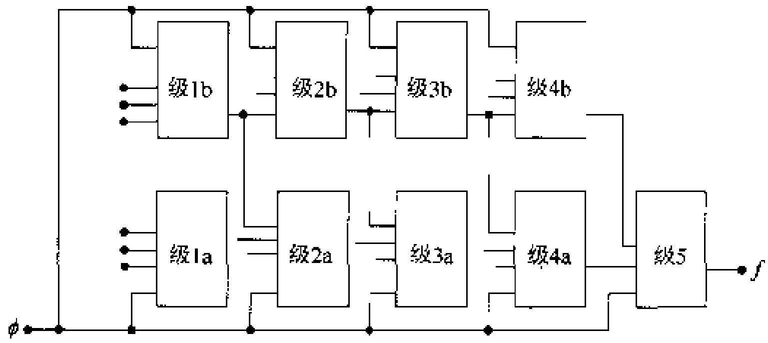


图 15.23 动态逻辑链

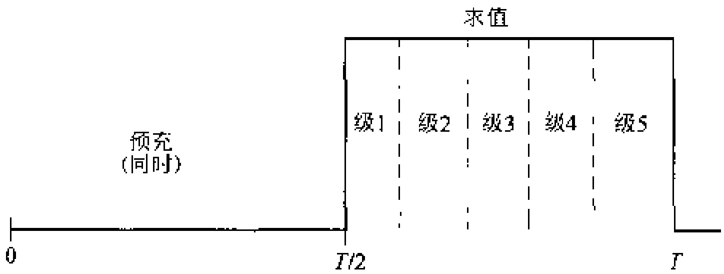
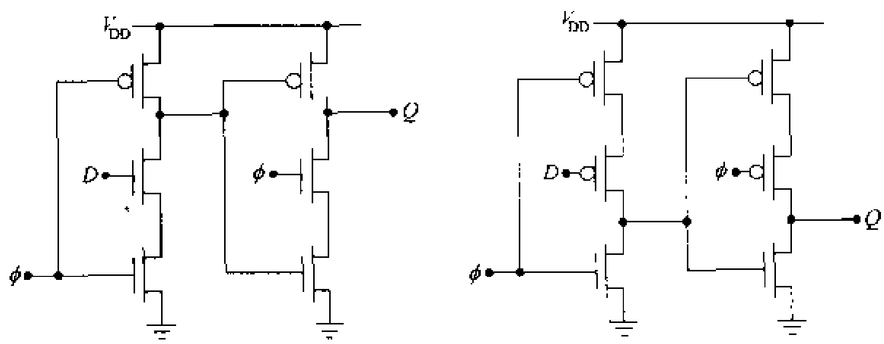


图 15.24 多米诺链的时序

这个例子表明数据传送进出一个动态逻辑链是由时钟来控制顺序的。每个时钟周期相应于这个逻辑链的一次完整的求值。在这个链中可能包括的级数是由每一级都切换情况下的延时来决定的。最大允许的求值时间由求值脉冲宽度($T/2$)确定。长的逻辑链可以采用相对较慢的时钟。然而这会引起电荷泄漏问题,因此必须有电荷保持电路。

尽管这个例子基于多米诺电路,但主要结果适用于大多数的动态 CMOS 逻辑系列。当采用动态逻辑链时,它们同时在输入和输出端处与静态电路接口。因此数据流是在系统一级形成的。

文献中发表了许多类型的动态 CMOS 锁存器。虽则大多数是单时钟两相位电路,但 TSPC(真单相位时钟)逻辑设计类型全部只采用单个时钟 ϕ。单相位的锁存器可以与静态门的设计接口以实现数据同步。图 15.25 为两个 TSPC 锁存器。图 15.25(a)中的“n 块”电路包括两级。第一级是一个简单的动态反相器,而第二级采用中间的钟控 nFET 实现锁存操作。预充电发生在当 $\phi=0$ 时,输出 Q 处于高阻状态(即为开路)。当 $\phi=1$ 时,第一级处于求值而



(a) nFET 实现锁存 (b) pFET 实现锁存

图 15.25 真单相位时钟锁存器

输出级的操作如同一个修改后的 NOT 电路。数据输入 D 被接受而一个经缓冲的值则出现在 Q 处。这个值在时钟返回 $\phi=0$ 时由 Q 点的输出电容保持。图 15.25(b)中的 p 块锁存器以类似的方式工作。TSPC 逻辑为 VLSI 设计提供许多吸引人的特点。然而因为它在本质上是动态性质的，所以会出现电荷泄漏和电荷分享的问题。

15.3 流水线系统

流水线是当一组顺序输入的不同数据，通过一个同步逻辑链时用来提高数据通过量的一种技术。因为计算机指令在本质上是顺序执行性质的，所以在微处理器中采用流水线来提高 MIPS 的数值。③

让我们通过分析图 15.26 简单寄存器 I/O 电路的时序要求，作为说明这些概念的基础。输入数据位 D_1在时钟上升沿时被锁存到 DFF 中，而在触发器延迟时间 t_{ff}之后则出现在 Q_1处。数据位 Q_1(连同其他没有显示的输入)进入组合逻辑电路且在延迟时间 t_d之后产生结果 D_2。在下一个时钟上升沿时，结果 D_2被锁存入输出 DFF。

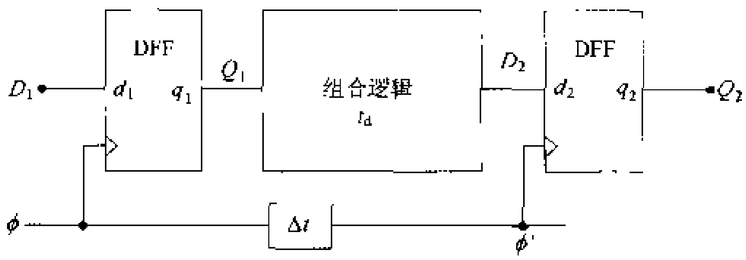


图 15.26 用于时序分析的基本流水线级

这个顺序可以建立对时钟波形的时序要求。因为数据在每个时钟上升沿处锁存入 FF，必须保证时钟周期 T 足够大以允许正常的电路延迟。图 15.27 为一组波形的例子。触发器延迟时间 t_{ff}及逻辑延迟时间 t_d分别显示在 Q_1和 D_2的波形上。两个 FF 时间显示在 D_1的图上。建立时间 t_{su}是在时钟边沿之前输入必须稳定的时间，而维持时间 t_{hold}是在时钟边沿之后输入必须保持稳定的最小时间，以便锁存正确的数值。我们介绍了可能出现的在输入时钟 ϕ 和输出寄存器时钟 ϕ'之间的偏差时间 t_s。这一组波形表明时钟周期 T 必须满足

$$T > t_{ff} + t_d + t_{su} + t_s \tag{15.18}$$

以满足所有的电路延时。维持时间的要求 t_{hold}对 DFF 的限制为：

$$t_{hold} < PW \tag{15.19}$$

式中 PW 是时钟的脉冲宽度。方程(15.18)使时钟频率限制为：

$$f < \frac{1}{t_{ff} + t_d + t_{su} + t_s} \tag{15.20}$$

在标准设计中，整个系统用单个时钟频率，所以系统的时钟是由最慢的子系统或单元决定的。

③ MIPS 是每秒百万条指令的缩写。

如果要求一个长的复杂的逻辑电路链，那么延迟时间 t_d将是决定这个电路时钟频率的关键因素。大的 t_d值要求长的周期从而减少了 f 的值。

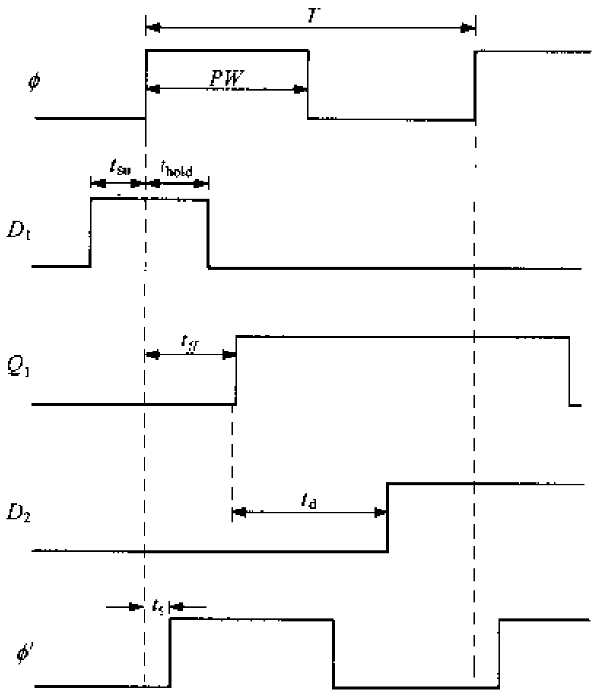


图 15.27 用于时序分析的波形参量

流水线系统设计用来提高一组顺序输入状态的整体通过量，它通过把逻辑链划分成较小的部分并且运用较快的系统时钟。图 15.28 显示了这个问题。输入寄存器把数据送入几条复杂的逻辑链，每条链的特征延迟时间为 t_d。具有最大 t_d值的链决定了这个单元的时钟频率；一条逻辑链的延迟时间数值是由电路设计类型、工艺参数以及物理设计确定的。

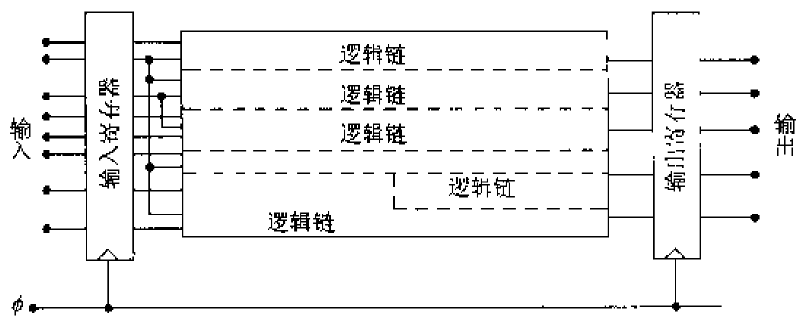


图 15.28 在时钟控制系统中的逻辑链

流水线的概念可以这样来理解，即注意到逻辑计算起始于输入部分，然后通过传播这条链到达输出端。一旦一个电路完成计算且把结果送到下一级，它就在这个时钟周期的余下部分闲置起来。图 15.29 显示了电路工作的进展情况，假设时钟的上升沿发生在时间 $t=0$。后续的时间表示为 t_i，这里 $t_{i+1}>t_i(i=1,2,3)$。图 15.30 的时钟波形表示相对的时间值。由于通过一个逻辑门的延时会因它的复杂性和寄生参数而有所不同，所以逻辑传播速率也将不一致，某些电路将比另一些电路有较长的延迟。

这个例子提供了流水线的基础。如果把长的逻辑划分成较小的组，在各部分之间加上寄

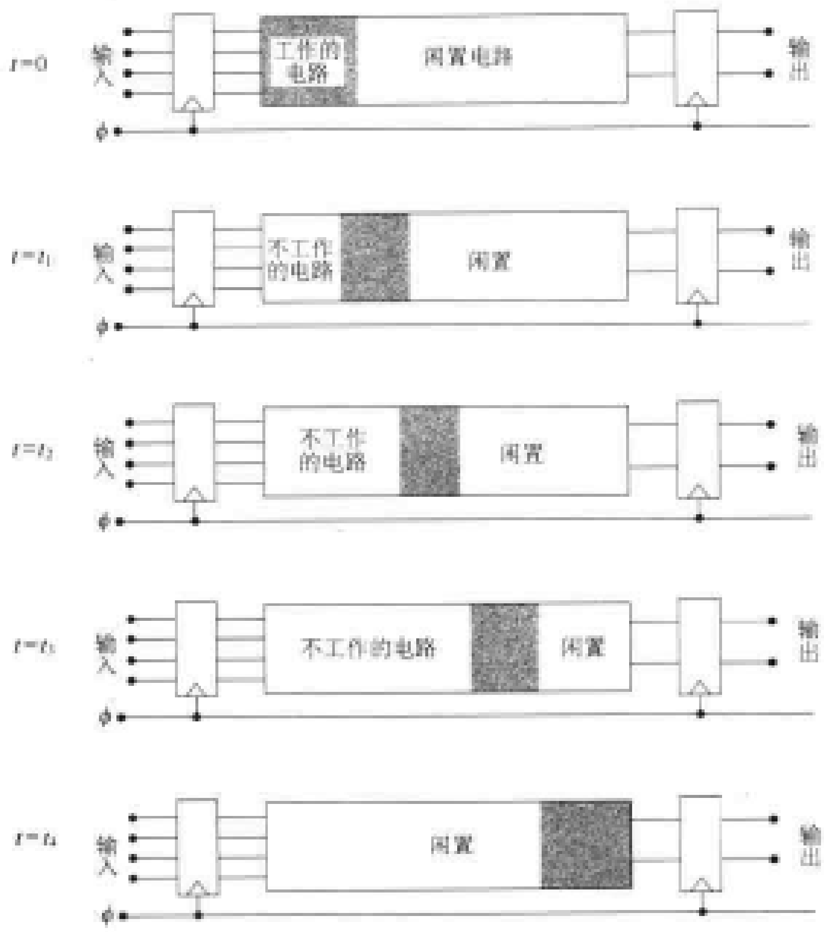


图 15.29 逻辑链中电路的工作情形

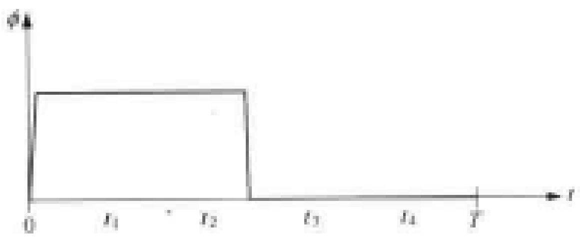


图 15.30 逻辑链中的时序进展

存器,并且运用一个较快的时钟,那么在任何给定时刻,大部分电路都在工作。图 15.31 显示了一个四级流水线。每级由一组输入寄存器和一个逻辑电路组成。时钟频率 $f_{pipe}=(1/T_{pipe})$ 由最慢的级确定并且要比原先逻辑链中使用的频率 f 高。然而,流水线设计是一种体系结构上的改变,它本身并不产生较快的响应。对于这个四级的例子,总的时间延迟将是 $4T_{pipe}$,而在非流水线电路中原先的时钟周期为 T。如果两个设计采用同一种 CMOS 工艺,那么有可能 $4T_{pipe}>T$。流水线吸引人之处是它增加了输出结果的速率。假设具有一组 N 个的顺序输入。在非流水线设计中,它需要总共 NT 秒来产生所有的结果。反过来,一条流水线一旦被充满之后则每个时钟周期产生一个输出(也就是每级同时在参与计算)。对于四级设计,产生整组输出所需要的总时间是:

$$4T_{pipe}+(N-1)T_{pipe}=(N+3)T_{pipe} \tag{15.21}$$

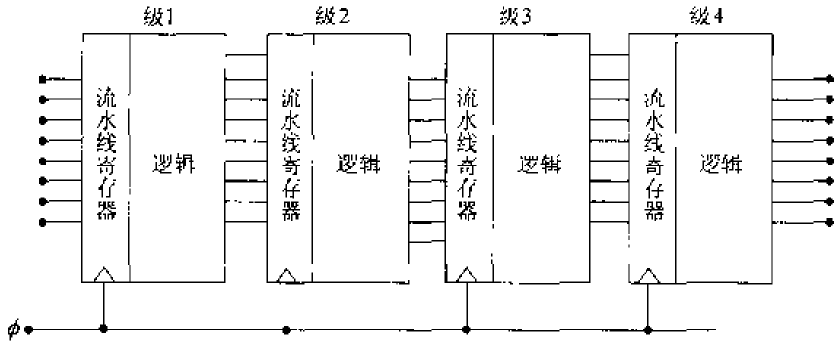


图 15.31 四级流水线

式中左边的第一项考虑了最初充满流水线所需要的时间。这是一个理想值并且假定保持一个充满的流水线。处理器数据通路中高层次的流水线设计是计算机体系结构中的一般问题，这里将不去讨论它的细节，④ 而去考察在 VLSI 设计中采用流水线的一些重要特点。

一个特别重要的问题是在原先的系统中选择切断点以定义较小的流水线级。对于第 i 级，最小允许的时钟周期 T_i是由以下条件决定的：

$$T_i > t_{ff} + t_{su} + t_{d,i} + t_{s,i+1} \tag{15.22}$$

如果在每一级都采用相同的流水线寄存器，那么($t_{ff}+t_{su}$)一项就是常数；同时时钟偏差时间 $t_{s,i+1}$是由时钟分配策略确定的。这就使逻辑延迟时间 $t_{d,i}$成为一个关键的因子。一旦选择了一条流水线级的逻辑行为与操作，逻辑与电路的设计就集中在使 $t_{d,i}$最小。对于一个 m 级的流水线，流水线的时钟周期选择为：

$$T_{pipe} = \max\{T_1, \ldots, T_m\} \tag{15.23}$$

因为它允许最慢的单元有足够的时间来完成计算。许多工程努力和时间都直接用来解决较慢的单元。已开发和实现了采用不同 CMOS 电路设计类型的各种算法解决办法。例如，在大多数电路为静态性质的芯片中，采用多米诺逻辑链设计关键的 ALU 电路。

提高速度的一个体系结构方面的方法是进一步划分一个或多个单元以增加流水线级的数目。这会降低通过一个单元的延迟时间，并且允许采用较快的时钟。然而，**深流水线**设计要求较多的时钟周期来完成整个链的计算，所以在时钟周期上的减小也许并不一定产生较快的电路。为使这个设计可以被接受，常要求相当大的工程努力。另一个影响因素是希望使每个流水线级相应于一个较大规模的行为功能。例如把一个存储单元分成较小的部分就会使设计和验证都困难得多。

物理设计考虑也是一个因素。芯片面积的成本及功耗的大小总是十分重要的。每个逻辑/电路的解决方案都具有不同的版图和工作特性，这些都必须在整个设计中考虑。流水线寄存器必须锁存每个输入位，因此在字长为 32 位或更多位的现代系统中它们的数量可能相当大。在芯片层次上，每个寄存器要求一个时钟信号及输入/输出线，它们使互连线的布线问题变得十分复杂。无论时钟还是信号的偏差问题都随版图面积而增加。尽管流水线是一个相当具体的结构设计类型，但所提及的大多数问题都会在任何先进的 VLSI 设计中遇到。

与基本流水线设计不同的另一种设计显示在图 15.32 中。它采用交替的正沿和负沿触发

④ 流水线结构的出色讨论包括在参考资料[8]中。

输入寄存器在时钟每次变化时锁存输入。这与仅采用正沿触发的触发器系统具有相同的基本问题。因为实际的限制仍然是通过逻辑电路的延时，因此并不能在提高速度方面获益，然而它确实允许采用易于产生和分配的较慢的时钟。

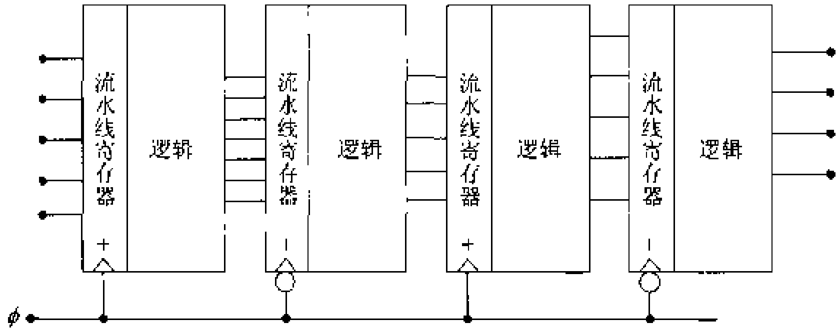


图 15.32 具有正沿和负沿触发的流水线

15.4 时钟的产生和分配

许多大系统的设计受限于时钟分配的问题。当时钟频率 f 达到 1GHz(10^9Hz)的水平即相应于时钟周期为

$$T = \frac{1}{f} = 1 \text{ ns} \tag{15.24}$$

时，就更成问题，因为这一时间已经接近门本身的开关速度。把时钟信号分配到芯片的各个点去是很复杂的，因为本征 RC 延时 τ 随线长 l 的平方增加，即：

$$\tau = Bl^2 \tag{15.25}$$

式中 B[秒/cm^2]是由物理版图和材料组成决定的常数。图 15.33 的芯片布局简单说明了这个问题。输入时钟信号 Clk 必须分配到每个单元。如果直接分配时钟，那么在线长度上的差别就意味着不同的信号延时。虽然每个单元将工作在相同的频率，但这些单元相互间在相位上就会不一致。所接收到的时钟 ϕ_a 将领先于 ϕ_b，而 ϕ_b 又会领先于 ϕ_c，如图 15.34 的波形所示。在本例中，ϕ_b 比 ϕ_a 延迟一段时间：

$$\Delta t_1 = B(l_b^2 - l_a^2) \tag{15.26}$$

而 ϕ_c 则比 ϕ_b 延迟

$$\Delta t_2 = B(l_c^2 - l_b^2) \tag{15.27}$$

同样的结论对于送入单元 D,E 和 F 的单线分配技术也是成立的。由信号分配线引起的时钟偏差问题是非常难处理的，特别是在大的芯片中。

芯片级的时钟通常分为两个范围的问题。第一个涉及电路设计，即从外加的参数时钟中产生时钟信号。第二个是考虑把时钟分配到片上的各个单元，使波形畸变和时钟偏差减到最小程度。这两个问题又是互相联系在一起的，即全局的时钟要求长的连线，而在时钟分配网络的一些地方又需要驱动电路。

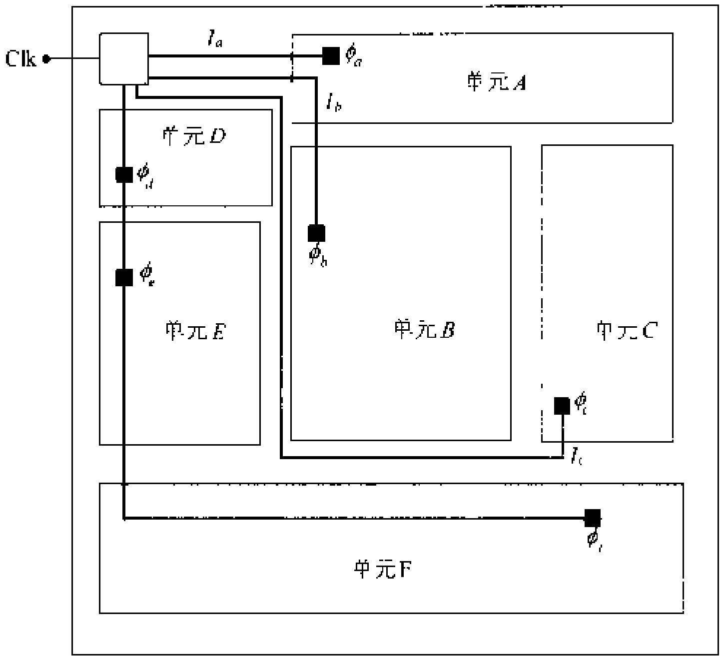


图 15.33　时钟分配到片内各模块

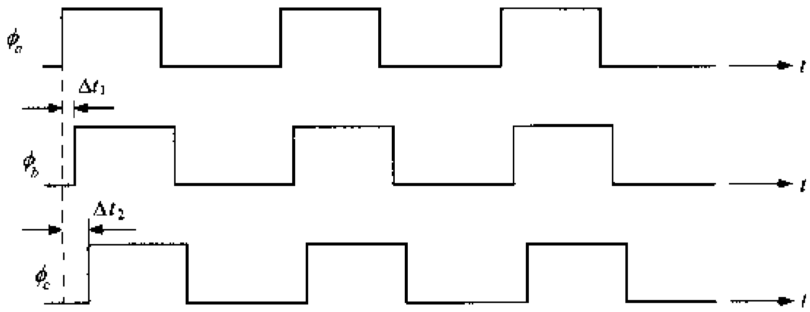


图 15.34　时钟偏差例子

15.4.1　时钟的稳定和产生

让我们首先考察**时钟稳定**的问题。一个外加的时钟信号控制在印刷电路板(PCB)上的系统操作。VLSI 芯片的内部电路必须与外部时钟同步。在高速电路中采用稳定电路,保证片内的计算能够和其他板级部件正确接口。

一个概念上简单的时钟稳定方法显示在图 15.35 中。外部的时钟信号加到**时钟**产生电路以产生必要的时钟波形。后者被送到一个电压控制的延迟线单元,在必要时可使这个信号变慢。一个逐级加大的缓冲器链提供把时钟分配到芯片所需要的驱动强度。这个电路的上面部分提供频率稳定。输出信号采样后送到相位检测电路以与外加的时钟进行比较。相位检测器产生的输出指明输出信号领先还是滞后于外部时钟。这个信息被低通滤波电路用来产生控制延迟线电路 RC 时间常数的电压 V_{adjust}。图 15.36 的**锁相环**(PLL)技术以类似的方式工作。PLL 设计来检测在输入和参考信号之间相位上的任何差别,并且产生与参考信号正确同步的输出波形。在这种情形中,外部时钟是输入信号,反馈回路则检查输出以进行修正。PLL 对稳定产生基波的高次谐波的频率乘法器电路也是很有用的。类似的电路也可以在串行数据传

输系统的**时钟恢复模块**中找到。时钟恢复电路从数据流中提取一位的时间间隔和时钟信号。在这类系统中，数据帧格式提供短的一串 0 和 1，它们提供同步的参照信号。

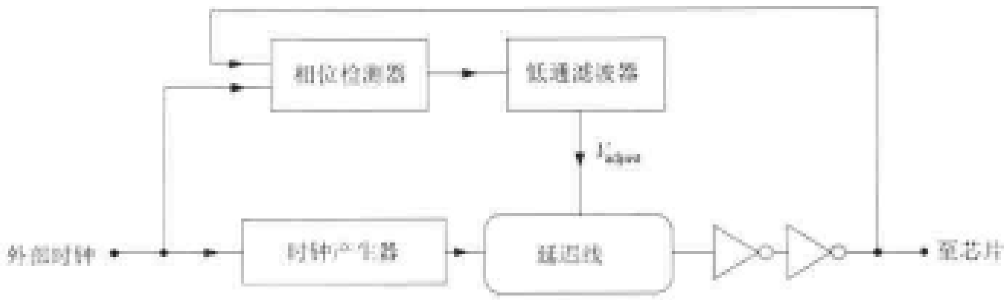


图 15.35　一个基本的时钟稳定电路



图 15.36　锁相环(PLL)时钟稳定电路

时钟发生器电路用比较简单的逻辑形式设计。使时钟电路的时钟偏差减到最小是通过仔细确定管子的尺寸来实现的。图 15.37 中简单的基于反相器链的电路从单个的输入 Clk 信号中产生互补的时钟 ϕ 和 $\bar{\phi}$。逐级加大的反相器链用来提供图中所示电容 C_1 和 C_2 所需要的驱动电流。这两个电容可以用如下的一般形式表示：

$$C = C_{line} + \sum C_G \qquad (15.28)$$

式中 C_{line} 是连线电容而第二项是与该连线相连的每个钟控 FET 栅电容的和。

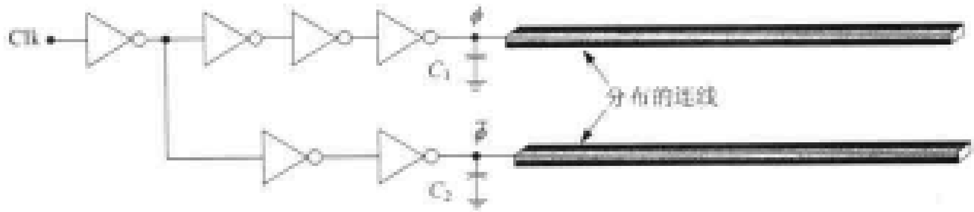


图 15.37　基于反相器的时钟产生电路

图 15.38 表示使时钟偏差最小的问题。图中可以看出分别通过上面和下面反相器链的 ϕ 和 $\bar{\phi}$ 两条路径。我们的想法是设计这两个电路使延时 t_1 和 t_2 相同。在逻辑努力(Logical Effort)技术中，这个电路由于它的形态[13]而称为**分叉电路**。设计基于使这两条链的电气努力(electrical effort)相同，但要注意 C_1 和 C_2(通常)并不相同。由此可以得到每条链中级的数目以及相对尺寸。

产生两个时钟相位的另一种技术是采用图 15.39 中的 D 锁存器电路。由于核心电路就是一个具有互补输入的 SR 锁存器，所以输出本身总是互补的且可用来作为 ϕ 和 $\bar{\phi}$。在输出波形之间存在两个时钟偏差的来源。较小的偏差可能来自反相器，而在输出电容 C_1 和 C_2 间的差别可以引起更多的问题。若有必要可以"拉长"这两个单个的 NOR 门来进行补偿。对这个

电路的修改显示在图 15.40 中,即增加反相器链来延迟反馈信号。它的作用是产生不重叠的输出,不重叠的间隔范围是由反相器链的延时决定的。

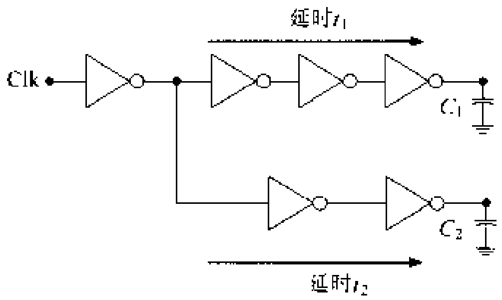


图 15.38 使时钟偏差最小的电路

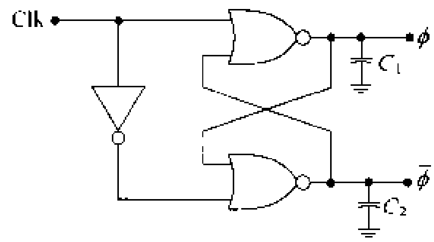


图 15.39 用锁存器产生互补时钟

15.4.2 时钟布线与驱动器树结构

一旦产生一组稳定的时钟信号,就必须把它分配到芯片上的子单元中去。在体系结构层次上,通常把时钟分配看成一个布线问题,其中各段连线的长度是很关键的。当实现大的电容和版图布线时,就要考虑物理设计问题。

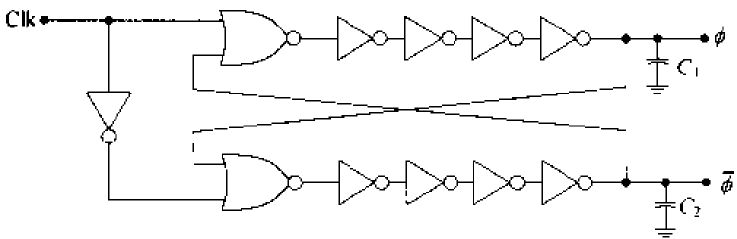
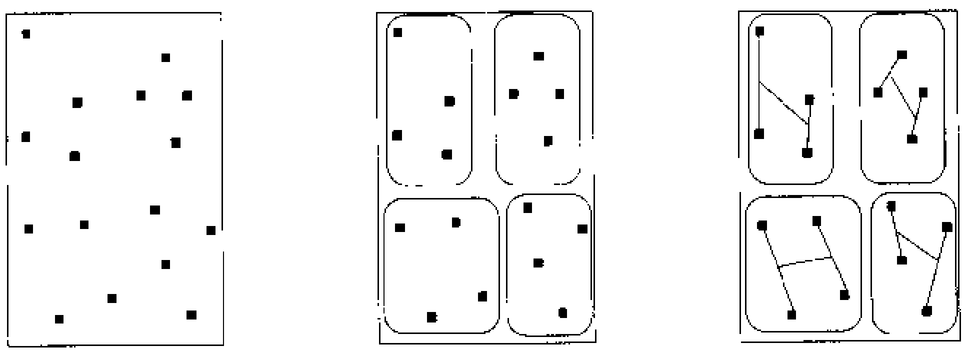


图 15.40 产生不重叠时钟的电路

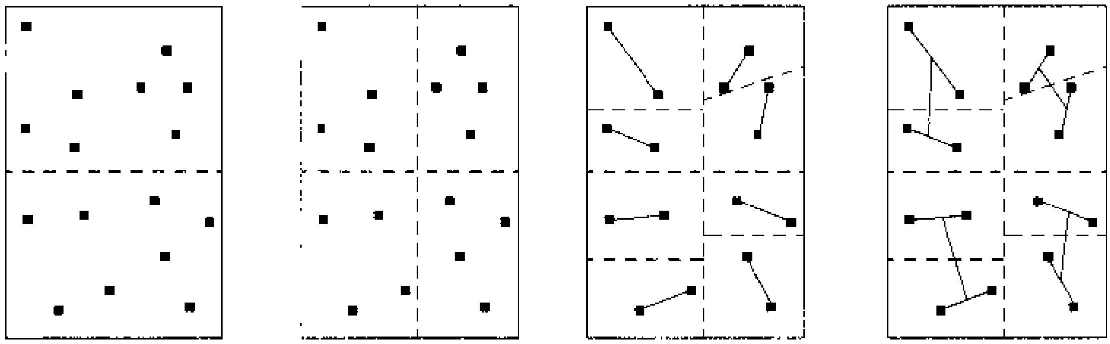
让我们首先来考察一下布线问题。假设最初的布局把时钟接收点按图 15.41(a)进行放置。因为它们是在整个芯片面积上分布的,所以把这些点组合在一起是很有用的;图 15.41(b)表示选择每个组含有 4 个接收点的情形。若把每个组内部的各个点用导线连在一起,问题就简化为驱动这些组的问题。图 15.41(c)显示了内部布线的一种方法,它用几乎是垂直的线来连接时钟接收点;这些垂直线用水平方位的导线连接起来。因为在本例中接收点的布置是随机的,所以不可能采用严格的垂直或水平布线。



(a) 时钟接收点 (b) 第一次分组 (c) 内部布线

图 15.41 时钟布线问题的简化描述

其他算法也可用来确定时钟分组。采用图 15.42 的分隔方法可以得到类似的结果。如图 15.42(a)所示,第一步分隔采用水平线。第二步分隔采用垂直线把芯片分成了 4 块,得到了如图 15.42(b)的分组。第三步基于水平边界线得到了如图 15.42(c)2 个点一组的结果;在组内部的各点可以按所示用导线连接起来。把两个两点一组连接起来就得到四点一组,于是就形成了图 15.42(d)中的最初布线。注意在这个例子中,分组和分隔技术得到的结果是很类似的。



(a) 第一步　(b) 第二步　(c) 第三步　(d) 布线

图 15.42 确定时钟分组的分隔步骤

上面的例子说明了几个重要的观点。甚至是中等复杂的设计也将具有几千个由时钟驱动的晶体管。为了简化布线问题,把位置接近的点组合在一起,然后集中在驱动各个组而不是单个的点。也许有必要构成"超级组",它由许多较小的组组成;在网络理论中,这与建立互连线布线树是相同的。这一过程就是起始于底部(各个接收点)并向顶部(时钟发生器)进行的层次化设计。一旦认识了这个问题的基础,就可以用自顶向下的方法找到解决办法。这意味着设计时钟分配策略时,首次通过在几何上定义时钟分组和时钟接收点。在完成之后,努力把这些单元适配到布局中以符合最优设计。

文献中研究了几种分配的几何形式。在高密度 VLSI 中最常用的策略之一就是 H 树。这一技术基于字母"H"的形状,并且确实很易理解。图 15.43 显示了一个"H",其中心点标以"X"。这一结构的对称性表明从中心到任何一端 A 的距离 l_{XA} 是一个常数。如果从 X 处发送一个信号且把接收点放在许多等同的 A 点处,那么延时

$$\tau_{XA} = Bl_{XA}^2 \tag{15.29}$$

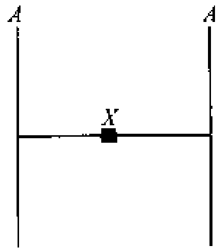


图 15.43 字母"H"的几何分析

都是相同的。换句话说,所有接收的信号都是同相位的。一个 H 树以 H 形状作为基本单元建立在几何格点上。图 15.44 显示一个宏设计。主时钟放在中心点 X,而接收的信号来自小的 H 的各个端点。这些点可以作为时钟源进行局部时钟分配,局部时钟则可以用更小的 H 树或直接用驱动器。虽然 H 树似乎是一个明显的解决办法,但必须记住以下两个条件应当满足:

- 每条时钟路径的长度和电气特性必须相同以产生所希望的效果;
- 在每个接收点处的负载电容必须相同。

因为树的对称性是很关键的,所以在分配树有最高优先权的地方通常有必要提供一层互连金属层。在构成树之后,其余的空间就可以按需要进行一般的布线。如果这一点不能做到,那么这树可以用不同的互连层来设计,但布线路径应当是相同的。以上第二个条件是属于电子设

计性质的。它要求在每一点处采用电气上相同的接收/驱动电路。

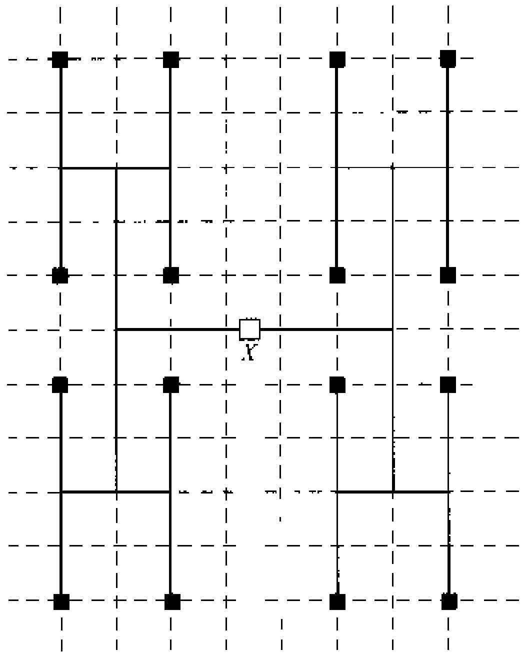
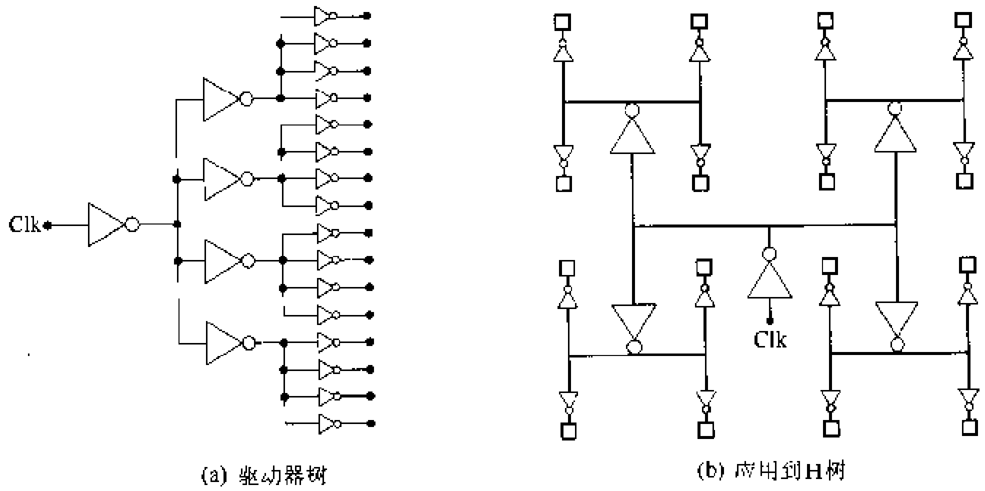


图 15.44 宏层次上 H 型的分配树

对理想的自顶向下设计的约束条件是必须安排布局和系统部件使能，包括由 H 树的几何位置确定的接收点。这在电路和物理设计层次上要求有不寻常的布局工作量，而且还可能无法实现。然而，所有这一切并不是白费的，因为主要的想法是使信号延时相同，而这是电子设计的问题。因此，通过仔细的电路设计仍然有可能与原来确切的要求有偏离。

一般可接受的观点是，控制时钟偏差最简单的方法是在分配路径中包括驱动器。如图 15.45(a)所示的**驱动器树**可把连线分成几段较短的长度并提供尺寸合理的缓冲器。驱动器的放置取决于分配策略；图 15.45(b)为一个 H 树的例子。驱动器树电路的设计基于反相器和连线的寄生参数。在图 15.46 中，采用了 π – RC 模型来包括连线电阻和电容。一旦每段的这些值已知，就可相应地确定驱动器的尺寸。这个模型表明分配网格的实际几何形态并不重要，因



(a) 驱动器树　　(b) 应用到 H 树

图 15.45 驱动器树的布置

为电气上的对称性可以通过使每组分支中 RC 部件的值相等来实现。可以构成许多可以接收的不同的 H 树。关键是在各对应部分保持连线的长度相同。图 15.47 所示例子中的布线方式提供到每个接收点相同的分配路径。尽管没有一个大规模的版图可以明显看出这一点，但版图确实提供这个希望的结果。这个灵活性在物理设计层次上特别是在高密度的版图设计中十分有用。

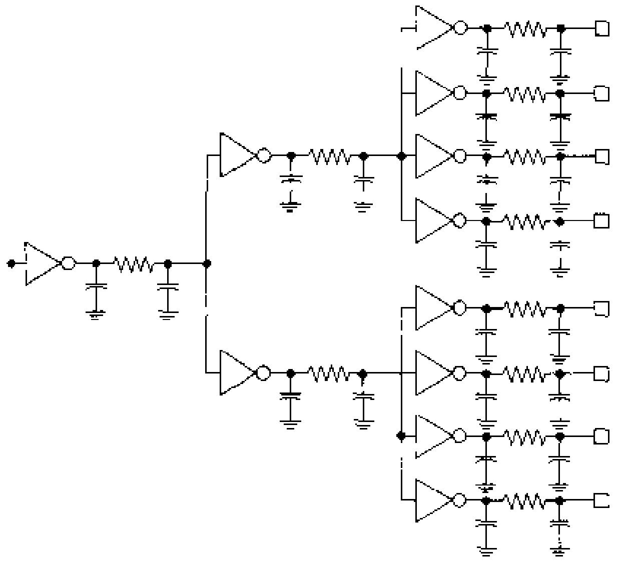


图 15.46　考虑互连线寄生参数时的驱动器树结构设计

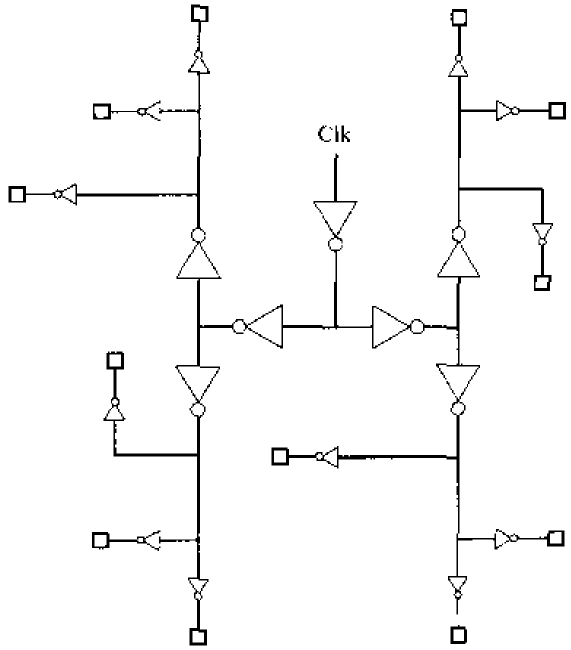


图 15.47　具有等同驱动器分段的分配技术

驱动器树的对称性同时简化了电气设计和版图设计。也可以构成一个具有可接收性能的非对称分布，但这是非常难以设计的。图 15.48 为一个采用非对称树分支结构的驱动网络。设计问题集中在运用公共的输入时钟在接收点 A，B 和 C 处产生对准(零偏差)的信号。理论上这可以在合理规定的偏差范围内实现。逻辑努力(Logical Effort)方法可以初步确定尺寸且对所得到的设计进行包括电阻的模拟。然而就实际意义而言，在设计周期中所附加的这些时间可能被认为是多余的。VLSI 依赖于在层次化设计中所有层次上的对称性和可重复性，并且应当适用于每种可能的情形。

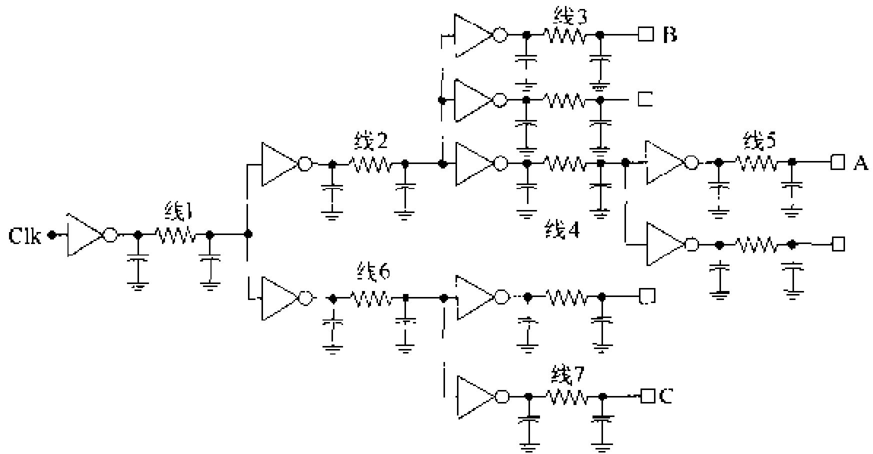
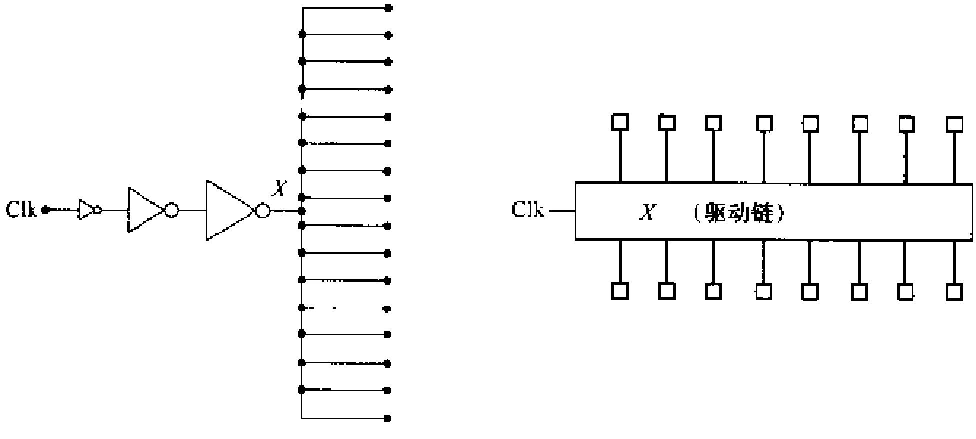


图 15.48　非对称分配的电路

放置时钟驱动器的另一种方法是把分配点看成电容负载的阵列，并且建立单个的缓冲器链来驱动整个组。图 15.49(a)为这种技术的等效电路。需要确定逐级增大的反相器尺寸以能驱动最后一级输出点 X 处看到的大电容 C_{out}。可以采用经典的尺寸缩放分析，决定级数及级比。因为 CMOS 电路必须建在硅层上，所以采用驱动器链比采用分配树要简单得多。因为在一个分配树中，必须连续不断地运用通孔和接触孔在硅(晶体管)和较高层的金属层(互连线)之间连接。采用单个驱动器树的布置图如图 15.49(b)所示。驱动电路形成一个大的驱动器"条"提供连接点；在实际中，这个"条"可以是输出级的大管子。分配线的长度决定了相对于 X 处相位的时钟偏差。全部都可以采用相同长度的线，或者也可以采用不同的长度有目的地形成对外部单元的延迟。



(a) 驱动器树 (b) 芯片的时钟分配

图 15.49 具有多个输出的单个驱动器树

采用各部分存在偏差延迟的方法提出一个有趣的观点。所有的时钟技术都假设希望同步系统中的每个单元，以使它们的相位相互一致。这可得到稳定和可靠的设计，但却并不总是易于实现的。如果不乐意采用这类系统时钟，那么也可采用另一种时钟技术。一种方法是采用**局部同步**的单元，即不同逻辑部分的工作相位与其他部分无关。这个概念显示在图 15.50 中。时钟 ϕ，ϕ' 和 ϕ'' 控制不同单元的操作。在 VLSI 层次上，它们可以具有相同的频率但却有不同

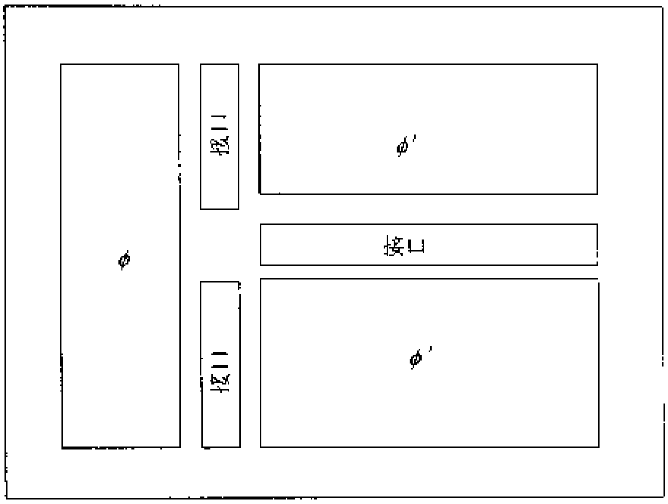


图 15.50 非同步系统时钟

的相位。在这些单元间的通信是通过接口逻辑电路实现的。

同步系统的操作可以采用较慢的全局时钟控制接口电路来实现。这种技术在板级设计中是熟知的。例如一个微处理器芯片的内部频率要比在个人计算机主板上用的系统时钟快得多。在当前的设计中,最快的 CPU 频率约为 1 GHz,而板级操作在最好的设计中大约为 100 MHz 至 200 MHz。时钟乘法器电路用来把板级频率提高到处理器的水平。单片 VLSI 设计的一个效率较高的方法是采用锁相环技术实现在单元间的通信。只要接口操作相对不频繁,那么这个技术就可以实现快速的系统操作。

在文献中引起广泛注意的一种非同步技术是**自定时**系统。一个自定时系统通常完全是非同步的,并且不运用任何外部提供的时钟。相反,一个自定时元件运用内部产生的信号对它的操作定时。图 15.51 为自定时元件的一个简单图示。系统接口是通过握手信号线来完成的,它们把一个请求信号送到这个元件以执行一个操作;当这个操作完成之后,这个元件以应答信号响应。应答信号可以送回控制单元,或送到下一个元件以构成一个自定时链。应答信号本身通常是内部产生的一个模拟延迟。自定时电路常用于许多应用中,如在 DRAM 阵列存取电路中。它们比较快并且一般要求较小的面积。

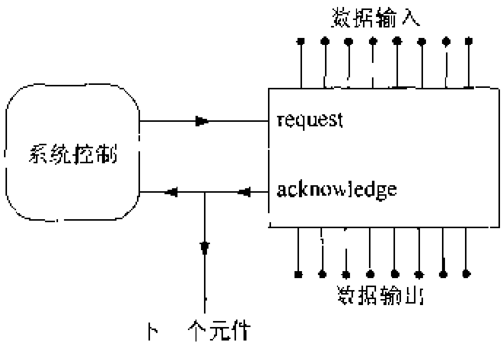


图 15.51 自定时元件的操作

理论上一个非同步机给出了最高性能,所以人们继续对使用自定时电路感兴趣。自定时 VLSI 系统的缺点是电路很难设计,并且电气操作对于因工艺波动引起的器件与电路参数的一般变化十分敏感。这使它很难完成一个能在常规生产线上大量生产的 VLSI 设计。在 VLSI 设计中采用自定时电路的一个实际问题是,没有及时开发合适的 CAD 工具处理自定时的从电路至系统的问题。

15.5 系统设计考虑

本书中一个重提的话题是 CMOS VLSI 设计不应当被看成只是另一种数字实现技术。尽管采用了高层次的结构模型来开始设计周期,但逻辑算法、电路设计类型以及物理设计诸方面都会影响最终产品的性能。记住这一看法是很重要的,因为它是开发下一代新体系结构和设计类型的关键。

VLSI 设计常常先对所有已知的相关因素权衡,然后基于满足最关键的要求而做出选择。然后再考察第二位的因素,以保证它们不会产生负面的影响。需要几个设计周期来确定所有的细节直到完成一个可以接收的设计。在 VLSI 方法中已经研究如何把逻辑电路从高层次的

抽象描述向下转化为物理设计层次，已经发现在每个层次上有许许多多的选择。然而有可能着重说明在过去几年中发展起来的 VLSI 设计的一些主要观念。我们将满足于把讨论局限在几个题目上。参考资料[12]阐述了设计整个微处理器芯片的一个出色的系统级的观点。

15.5.1 位片式设计

考虑一个运算和逻辑单元(ALU)，它接受 n 位的字 A 和 B 并产生一个 n 位的结果 C，如图 15.52 所示。在体系结构层次上，ALU 可以描述如下：

$$C = C(A, B; \text{control}) \tag{15.30}$$

其中控制信号决定了 C 的功能与输入 A 和 B 的关系。可以采用图 15.53 所示的方块图来描述 ALU 的操作。在这张图上，A 和 B 输入被送到大逻辑块内部不同的逻辑部分中去。每个部分接受 n 位的字作为输入，但产生一个不同的 n 位输出 f_k，这里 $k = 0, 1, \cdots, (m-1)$，m 是逻辑函数的数目。所有这些结果都计算出来，但 n 位的 ALU 输出 C 是由加在 m:1 的 MUX 上的控制信号决定的。

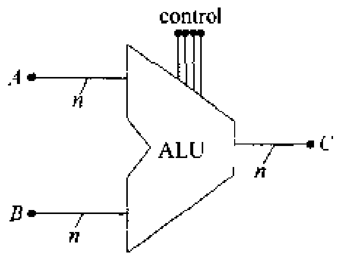


图 15.52 一个 n 位的 ALU

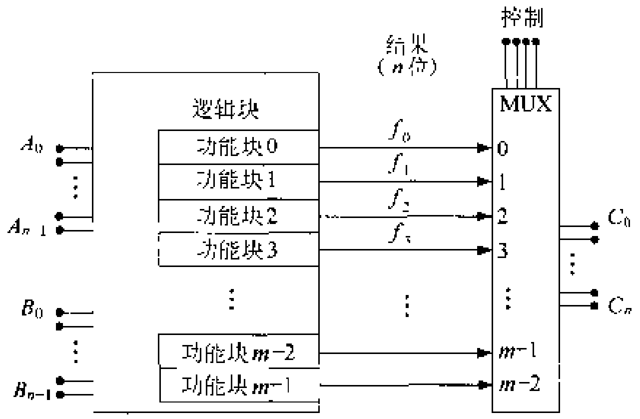


图 15.53 ALU 的方块图

位片式设计的基础是逻辑针对位一级的数据。字尺寸的操作采用一个个平行连接的位级电路，它们执行相同的操作。图 15.54 是图 15.53 整体方块图中第 p 位的位级方块图。输入为 A_p 和 B_p，p 在 $[0, n-1]$ 内；输出 C_p 为结果。采用位片式原理，一个 n 位 ALU 的实现可以通过把 n 个相同的单元并行起来，如图 15.55 所示。注意，在方块图中看到的单元重复情形

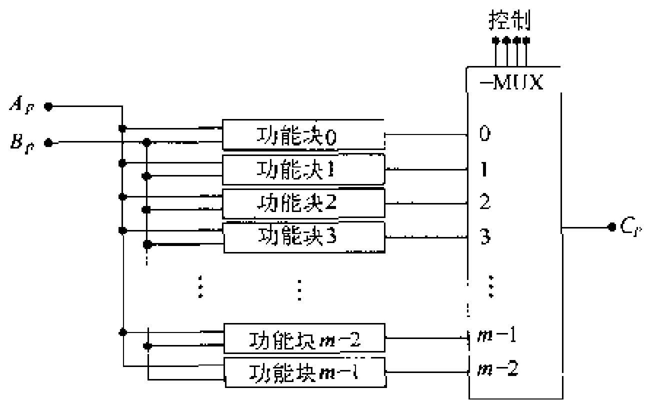


图 15.54 一个 ALU 的位片结构

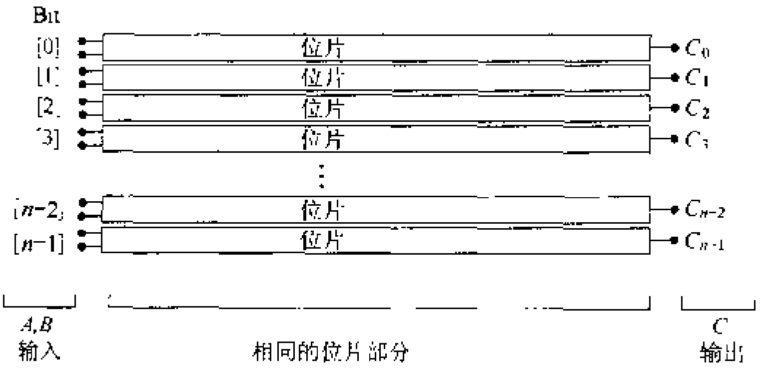


图 15.55 用位片式设计 ALU

也将出现在逻辑、电路和硅片层次上。这种方法的优点是一旦设计出位片，就可把它放进一个库中，需要时就可以调用出来，构成所希望的 ALU 宽度。这类似于在 Verilog 中建立一组一位的数据类型：

```
reg A,B,C;
```

然后再把它扩展成一个 32 位的字，即把它改写成如下的形式：

```
reg[31,0]A,B,C;
```

由于这一特点，位片式设计的功能非常强。

在 VLSI 层次上，经验表明采用位片式概念可能改进也可能不改进设计。在单个位片版图中各条互连线的布线简单得多，而且单单这一因素就使得这个技术十分吸引人。然而有些电路，例如 CLA 电路，最好设计成 4 位或更多位输入的电路。这是因为这一算法本身能够同时接收输入字的好几位。把这些字划分成单个位的电路将使相邻位片间的布线变得复杂。在这些情形中，也许以相同的方式采用多位的单元更为有效。另一种方法是在数据通路中采用不同的单元宽度。

在物理设计层次上遇到的一个问题是位片建立在硅片上时的尺寸和形状。这些位片需要安装在一起，形成一个模块放到布局中去，但这常常会使各个位片具有不成比例的单元宽高比即宽:高。如果一个比另一个大很多，那么长互连线可能成为一个问题。反过来，如果宽:高约为 1:1(这表明是一个正方形)，那么有可能很难去构成一个字单元具有布局方便的形状。

这个例子说明了把一个简单的体系结构概念应用到 VLSI 设计中时会产生的一些问题。随着在设计层次中更向下一步，这些问题仍然是相同的，但对它们的观点和结果却会有所改变。

15.5.2 cache 存储器

在微处理器设计中，较大的系统存储器放在系统主板上，并且在物理位置上和电气上都远离中央处理单元(CPU)。同时系统板级的时钟要比 CPU 内部的时钟慢得多，所以访问主存通常被认为是在数据通路中最慢的操作之一。

cache 存储器设计放在 CPU 和主存之间以加速系统的操作。cache 由小部分的快速存储器(SRAM)构成，它们可被处理器作为本地的读/写存储器。它与系统存储器通信并且允许传送数据块。图 15.56 表示了这一概念。图 15.56(a)的存储器直接存取可能很慢。如图 15.56(b)增加片内 cache 可以使系统保持芯片级的速度。在 cache 层次中最近的 cache 称为 L1(第

一层)cache;它通常是一个片上阵列。[5] 该图显示了在 CPU 设计中采用的两类本地存储器。I-cache 用来保持指令,指令从存放程序代码的主存储器中取出。保持 I-cache 尽可能地装满,允许程序自由运行。数据 cache(D-cache)保持计算结果;这些结果可以传送到主存储器或保持作为后续指令的操作数。

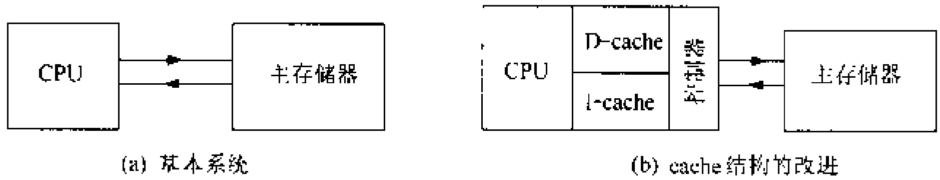


图 15.56 增加 cache 存储器

I-cache 和 D-cache 单元用在超标量计算机结构中,后者采用多条流水线数据通路以提高指令处理速度。图 15.57 为双发射超标量设计的一个简单方块图。指令在 I-cache 中存放和分类,并且按照必要的顺序被送往其中一条流水线。结果则累积在 D-cache 中,根据需要使用。cache 控制器电路把数据流导入或导出主储存器,它必须保持主存储器能用新的结果来更新并且保持指令 cache 尽可能地装满。在系统层次上有关 cache 和超标量设计的见解,在关于计算机结构的许多教科书中都有讨论,这里将不包括这一内容。

在 VLSI 设计中,对 cache 的需要基于物理限制,它们影响系统的性能。不可能在芯片上提供大量的存储器,因为它需要较大的面积。可以设想在任何输入/输出端口速度都会变慢,所以非常希望有片内存储器。cache 单元的尺寸不可能太大。因为它是为了提高速度而加入系统中的,所以采用了 SRAM 电路,尽管它们比 DRAM 单元消耗更多的面积。库的设计非常有用,因为它们可以直接被调用到芯片设计中并且它们的特性也被很好地描述。

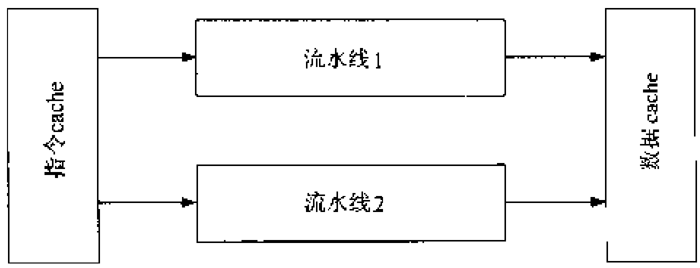


图 15.57 双发射超标量机的方块图

15.5.3 脉动系统和并行处理

系统级的 VLSI 可以看到通过由许多部件构成的大系统的数据流。在脉动系统中,数据的移动由时钟控制,每周期移动一个相位。其名字类似于心脏泵动血液。脉动设计对在数字信号处理(DSP)中各种类型算法的硬件实现是很理想的。因为 DSP 是按离散时间尺度来处理事件的,一个时钟的每个周期启动新的一组计算。

并行处理的领域涉及类似的体系结构。它的目的是设计一个包括几个各自连接在一起的**处理器单元**(PE)的计算系统。一个 PE 可以简单到一个 AND 门,或可以复杂到像一个通用的

⑤ 层次 2(L2)cache 是在 L1 cache 之外的下一个层次。

微处理器。一个并行机是设计来使许多 PE 同时工作在一个程序的不同部分。在 VLSI 实现中,PE 的作用如同一个单元,它可以通过例举来复制。图 15.58 显示并行处理网络具有的基本特征。各个处理器单元通过开关阵列进行通信,后者由本地的或全局的信号控制。通信路径由硬布线开关和控制网络导向。每个基本单元在整个系统中重复,因此利用了 VLSI 最强的特点之一。建立这类巨型处理器的欲望促进了实现圆片集成(WSI)设计的努力。由于在制造无缺陷圆片中涉及到的困难,这还没有能够实现。

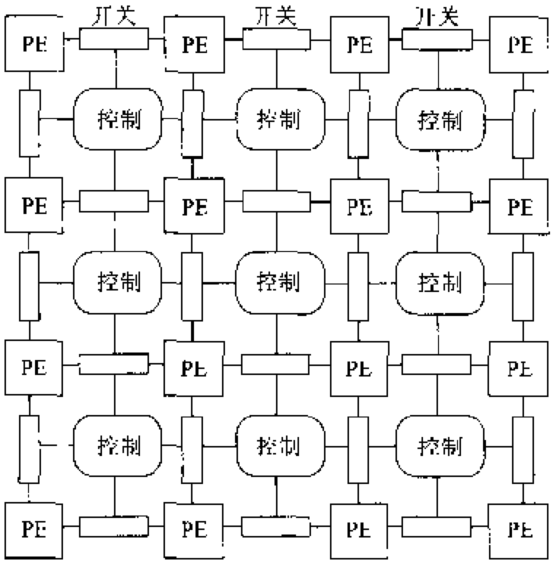


图 15.58 并行处理网络中的规则排布

15.5.4 小结

讨论这些简短的例子是为了强调在设计层次内部不同层次间的相互作用,必须进行综合考虑以保证所完成的产品具有所希望的特性。在一个层次上改变设计通常会要求在大多数其他层次上改变设计。自然这是通贯全书的观点。

VLSI 系统设计是一个富有挑战性的领域,这是由于它所产生问题的数目和类型决定的。新的产品总会使人惊奇,特别是芯片出自于当代最高技术水平的考虑。

15.6 参考资料

[1] Abdellatif Bellaouar and Mohamed I. Elmasry, **Low-Power Digital VLSI Design**, Kluwer Academic Publishers, Norwell, MA, 1995.

[2] H. B., Bakoglu, **Circuits, Interconnections, and Packaging for VLSI**, Addison-Wesley, Reading, MA, 1990.

[3] Kerry Bernstein, et al., **High Speed CMOS Design Styles**, Kluwer Academic Publishers, Norwell, MA, 1998.

[4] Stephen D. Brown, **Field-Programmable Gate Arrays**, Kluwer Academic Publishers, Norwell, MA, 1992.

[5] William F. Egan, **Phase-Lock Basics**, Wiley-Interscience, New York, 1998.

[6] James M. Feldman and Charles T. Retter, **Computer Architecture**, McGraw-Hill, New York, 1994.

[7] John P. Hayes, **Digital Logic Design**, Addison-Wesley, Reading, MA, 1993.

[8] David A. Patterson and John L. Hennessy, **Computer Organization & Design**, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 1998.

[9] Jan M. Rabaey, **Digital Integrated Circuits**, Prentice-Hall, Upper Saddle River, NJ, 1996.

[10] M. Sarrafzadeh and C. K. Wong, **An Introduction to VLSI Physical Design**, McGraw-Hill, New York, 1996.

[11] Navid Sherwani, **Algorithms for VLSI Physical Design Automation**, Kluwer Academic Press, Norwell, MA, 1993.

[12] Bruce Shriver and Bennett Smith, **The Anatomy of a High-Performance Microprocessor**, IEEE Computer Society Press, Los Alamitos, CA, 1998.

[13] Ivan Sutherland, Bob Sproul, and David Harris, **Logical Effort**, Morgan Kaufmann Publishers, San Francisco, 1999.

[14] John P. Uyemura, **A First Course in Digital Systems Design**, Brooks-Cole Publishers, Pacific Grove, CA, 2000.

[15] John P. Uyemura, **CMOS Logic Circuit Design**, Kluwer Academic Publishers, Norwell, MA, 1999.

第16章 VLSI电路的可靠性与测试

VLSI测试技术用来决定在制造顺序完成之后一个芯片的性能是否正确。如果一个芯片通过测试阶段，则进行封装并且可以销售。**可靠性**是有关预测一个部件一旦开始使用后它的寿命情况。

16.1 一般概念

让我们首先考察一旦生产阶段完成圆片会出现什么情况。生产进行了几个星期的最终结果是圆片上排成阵列的芯片。每个芯片都可能是一个工作正确的电路，它们可以被封装和销售。可惜的是由于在生产过程中出现的问题或随机的变化，并不是每个电路都能像所设计的那样工作。每个芯片都要经过一连串经选择的电气测试以决定电路是好的还是坏的。

圆片的测试步骤示意在图16.1中。一个**测试探针**与一个芯片的I/O点进行电接触。几组激励被加到输入端而响应从输出端取出。测试系统设置成根据这一组测试的结果来接受或不接受这一芯片。一个坏的芯片被做上标记以供以后参考。每个芯片都测试之后，在圆片上沿着每个芯片位置之间的走线划开。对圆片施加一点压力就会使它沿划片线断开，于是就得到了一个个的芯片而不损坏电路。好的电路进入最终组装阶段，此时利用机器人设备把芯片放进管壳中，连接芯片与管壳的电极，然后把管壳封起来。

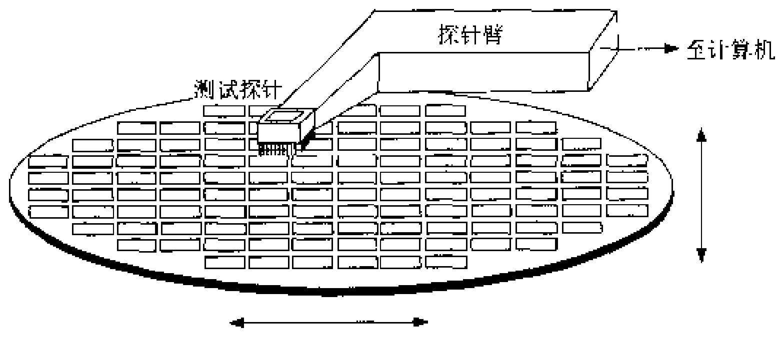


图16.1 圆片测试示意图

一旦一个集成电路用在板级设计中时，芯片的可靠性就成为一个重要的因素。每个电子电路最终都会“用尽”，VLSI芯片也不例外。预测**寿命**是在失效发生之前期望的工作小时数。这是用图16.2所示的**浴盆曲线**① 来表示的，它画出一个给定系统失效次数与时间的关系。浴盆曲线是半对数坐标图，其中时间 t 的单位为小时，而曲线值是按 $\lg(t)$ 画出的。在曲线中

① “浴盆曲线”的名字来源于可靠性曲线的形状。

显示了三个基本的区域。**"早期失效"**是在非常短的时间之后发生的失效，也就是发生在系统寿命早期的失效。这些多半是由于制造缺陷，在工作几个小时之后这些缺陷本身就变得明显了。曲线的中心部分代表在正常工作期间的**随机失效**，而**"用尽"**则描述了寿命的末端。

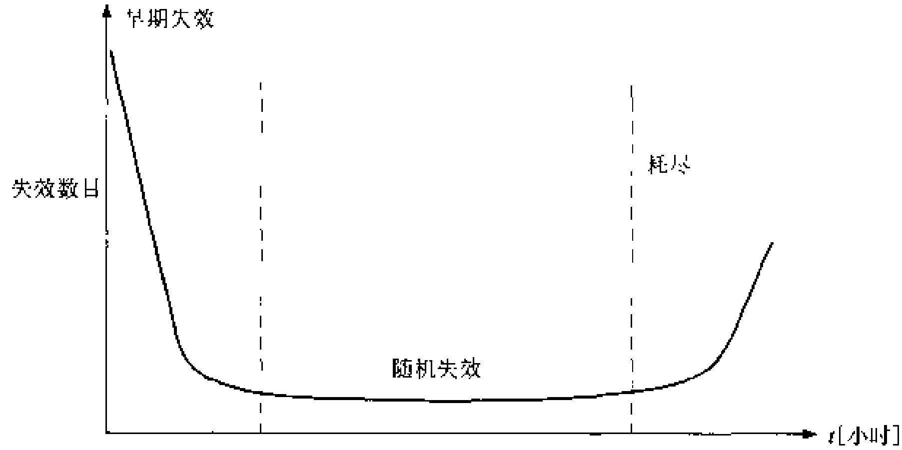


图 16.2 浴盆形状的可靠性曲线

早期失效的数目对任何器件都可能是很大的，特别在设计或制造正在改进的头几个月生产期间。一旦一个电路被安装在板上，修复的成本要超出 IC 本身的成本，所以希望通过**老化**操作把尽可能多的早期失效情况排除出去。在老化过程中，电路是在强化的条件下工作的，即在比正常电压高的电压下，在高温下，以及在高湿度环境下进行。其想法是把潜在的失效引发在老化期间发生从而避免在板级上运用这些器件。这相应地提高了系统的可靠性。电子部件的供应商，包括 VLSI 芯片的制造商，通常对他们的产品都提供某些形式的书面担保。因为生产可靠的元件对两者都是有益的。

老化系统也用于获得可靠性数据。一大批电路样品被通电并放在炉子或潮湿的小室中，监测这组测试样品的性能降低和/或失效，且对每种情形记下发生的时间。一年只有 24×365＝8760 个小时，所以某些测试系统会连续工作好几年！一种技术是测量关键的参数(或一组参数)并且跟踪性能随时间降低的情况。图 16.3 显示的理想化例子是一个电路的电源电流 I 发现为随时间增加的情形。最大允许的电流是用 Max 表示的。在规定时间间隔取数得到的数据点显示这个测试组的高值、低值和平均值。这可以用来外推期望的寿命时间。最简单的直线法采用所示的虚线来近似电流。把它外推到 Max 值就得到了对这个器件寿命的估计。采用这种方法的一个问题是由于时间轴是对数坐标，所以这条直线斜率的稍微改变会明显改

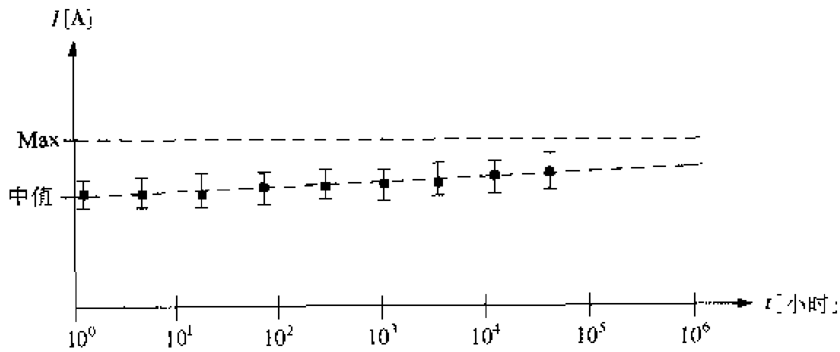


图 16.3 测试数据结果的例子

变所预测的寿命。

可靠性建模

可靠性数据可以构成对预测失效率很有用的数学模型。假设从时间 $t=0$ 开始测试一组器件。随时间推进,一个器件在时间 t_1时失效,然后下一个在时间 t_2时失效,依次类推。在测试周期结束时,发现 N 个器件失效。这组器件工作时间的总数为以下的和:

$$T = \sum_{i=1}^{N} t_i \tag{16.1}$$

将以小时来度量它。据此可以定义这个测试组的**平均失效率**,它是失效次数除以工作小时的总数:

$$\lambda_{av} = \frac{N}{T} \tag{16.2}$$

平均失效时间(MTTF)由下式给出

$$\text{MTTF} = \frac{T}{N} = \frac{1}{\lambda_{av}} \tag{16.3}$$

它代表这个测试组的平均寿命。

事实上,若用小时为单位来描述失效率,它的值会非常小。一个比较常用的度量单位之一是 **FIT**。它代表**在一固定时间内的失效次数**。即它定义为

$$1\ \text{FIT} = \text{在1000小时内100万个部件失效一次} \tag{16.4}$$

或

$$1\ \text{FIT} = 1\times10^{-6}/\text{K} \tag{16.5}$$

1FIT 的失效率相当于大量采样中的一个器件它的寿命约为 125 000 年(10^9工作小时)。所谓**强化寿命测试**是采用极端的条件来仿真器件的老化,它常常用来决定合理的 FIT 数值。

【例 16.1】 考虑一个具有 200 000FET 的小芯片。FIT 需要什么值才能达到平均可靠性为一年中不超过 1 个管子失效?

假设每年为 8760 小时,这些 FET 管代表总共为$(200\,000)\times(8\,760)=1.752\times10^9$器件·小时/年。为求出达到每年 1 个失效所需要的 FIT,可以写出:

$$\left(\frac{x}{10^9}\right)(1.752\times10^9) = 1 \tag{16.6}$$

式中,x 是 FIT 值。求解得到 $x=0.67\text{FIT}$ 就是所要求的失效率。

这个简单的失效分析方法提供了很有价值的观察,但并不足以产生精确的估计。比较成熟的数学模型提供更高的可信度,因而用在整个领域中。为了建立这个一般的方法,假设度量在一个测试组中的失效并且发现它是时间 t 的函数。可以引入时间 t 的函数即**概率密度函数**(PDF):

$$f(t)\,dt = \text{在时间增量}dt\text{期间的失效次数} \tag{16.7}$$

一个简单的模型是指数函数

$$f(t) = \lambda e^{-\lambda t} \tag{16.8}$$

式中 λ 是常数。**累积分布函数**(CDF)$F(t)$与 PDF 的关系为

$$F(t) = \int_0^t f(\eta)d\eta \tag{16.9}$$

式中 η 是一个名义积分变量。对于指数分布,可对方程(16.8)积分得到

$$F(t) = 1 - e^{-\lambda t} \tag{16.10}$$

器件寿命是用称为**寿命分布**的模型来描述的。CDF 是寿命分布,它可以被解释为:

- $F(t)$是测试组中一个随机单元将在 t 小时内失效的概率;
- $F(t)$是测试组中在 t 小时内失效的单元数目。

这两种解释在实际中都是很有用的。**可靠性函数** $R(t)$定义为

$$R(t) = 1 - F(t) \tag{16.11}$$

它描述没有失效的单元,也就是在 t 小时之后仍然工作的单元。

这些函数可以定义**失效概率密度** $h(t)$为

$$h(t) = \frac{1}{R(t)}\left(\frac{dF}{dt}\right) \tag{16.12}$$

它可以解释为在时间 t 和$(t+dt)$之间一个单元失效的概率。求这个导数值得到

$$h(t) = \frac{f(t)}{R(t)} \tag{16.13}$$

失效概率密度在文献中又称**即时失效率**或**称失效率**。② 图 16.2 的浴盆曲线就是失效概率密度的例子。将失效概率密度积分就得到了**累积失效函数**

$$H(t) = \int_0^t h(\eta)d\eta \tag{16.14}$$

可以证明它等于

$$H(t) = -\ln[R(t)] \tag{16.15}$$

于是得到

$$F(t) = 1 - e^{-H(t)} \tag{16.16}$$

即为 $F(t)$和 $H(t)$之间的关系。

在两个时间 $t_2 > t_1$之间的**平均失效率**(AFR)为累积的失效率与这个时间间隔的比。假设为简单起见,$t_2 = T$ 及 $t_1 = 0$,那么

$$\mathrm{AFR}(T) = \frac{1}{T}\int_0^T h(\eta)d\eta \tag{16.17}$$

它可以用另外一种形式来表示:

---

② 我们注意到某些处理方法定义失效率为 $f(t)$而不除以 $R(t)$。

$$\mathrm{AFR}(T)=\frac{H(T)}{T}=\frac{\ln R(T)}{T} \tag{16.18}$$

AFR 在说明一个器件工作 T 小时的失效率时非常有用。

让我们考察具有指数分布的这些函数的应用。假设对芯片测试组进行一次强化寿命测试实验并且画出失效的数目，得到图 16.4(a)所示的直方图。可用连续分布模拟这一数据：

$$f(t)=\lambda \mathrm{e}^{-\lambda t} \tag{16.19}$$

式中 λ 的确切值可以从曲线中估计。这一曲线显示在图 16.4(b)中。由于

$$F(t)=1-\mathrm{e}^{-\lambda t} \tag{16.20}$$

所以失效率为：

$$h(t)=\frac{1}{R(t)}\left(\frac{\mathrm{d}F}{\mathrm{d}t}\right)=\lambda \tag{16.21}$$

也就是，失效率是不随时间变化的常数，其值为 λ 失效率/小时。这是由指数函数提供的简单模型的特性。平均失效率可以计算为：

$$\mathrm{AFR}(T)=\frac{\lambda T}{T}=\lambda \tag{16.22}$$

它也是一个常数。一般地 MTTF 由以下的"矩"(moment)给出：

$$\mathrm{MTTF}=\int_0^{\infty} t f(t)\mathrm{d}t \tag{16.23}$$

对于指数分布，通过分部积分法得到：

$$\mathrm{MTTF}=\int_0^{\infty} t\lambda \mathrm{e}^{-\lambda t}\mathrm{d}t=\frac{1}{\lambda} \tag{16.24}$$

由指数分布得到的结果与采用直观写出的较简单的表达式是相同的。这一分析的限制条件是假设失效率为常数。

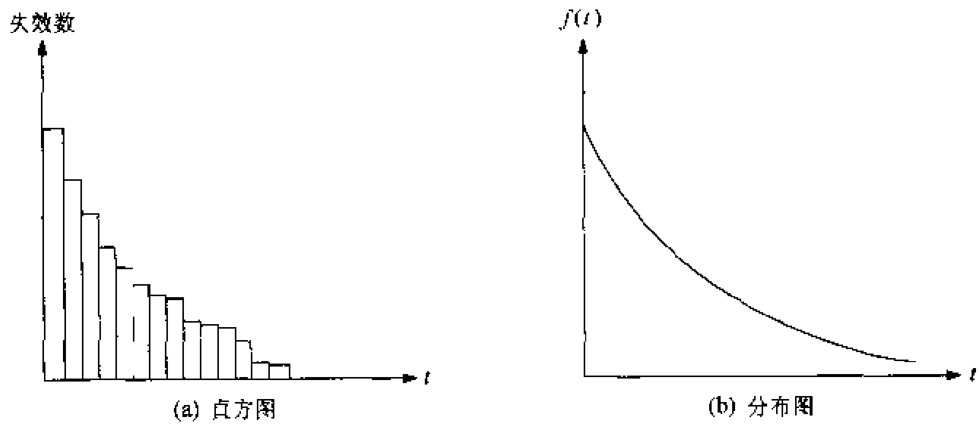


图 16.4　指数失效模型

不同的分布函数 $f(t)$ 可以用来模拟非常数失效率的情形。**Weibull 分布**应用

$$f(t)=\frac{m}{t}\left(\frac{t}{c}\right)^{m}\mathrm{e}^{-(t/c)^{m}} \tag{16.25}$$

式中，$m>0$，$c(>0)$称为**特征寿命**。由此得到失效率为

$$h(t)=\frac{m}{t}\left(\frac{t}{c}\right)^{m} \qquad (16.26)$$

及平均失效率为

$$\mathrm{AFR}(T)=\frac{1}{c}\left(\frac{T}{c}\right)^{m-1} \qquad (16.27)$$

分布的形状可以通过调整 m 来选择。m 为奇数值则产生类似于指数的曲线，而 m 为偶数则得到比较类似于高斯的分布，其峰值在某个 c 值处。例如，$3\leqslant m\leqslant 4$ 就产生钟形曲线。其他分布，包括正态和对数正态(log-normanl)函数，也用在可靠性建模中。

可靠性建模是一个令人非常感兴趣的研究领域，它采用数据统计分析力图决定预测的寿命和失效率。一个特别富有挑战性的问题是设计实验及相关的模型以提供有意义的数据。随着 VLSI 工艺设备复杂性的提高，可靠性问题变得更加关键。许多具有很强物理学和统计学建模背景的工程师和科学家正致力于这个领域的研究。

16.2 CMOS 测试

让我们把讨论转到测试数字 CMOS 电路的问题。它的概念非常简单。给定一个数字集成电路，希望决定它是否正确工作。假设逻辑是正确的并且应当有可能制造出一个工作的芯片。因为功能是由设计定义的，可以算出这个电路对一组输入激励应当如何做出响应并且采用这个信息去进行实际的测试。

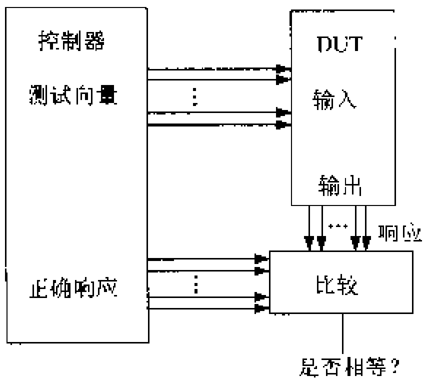


图 16.5　测试问题略图

测试的一般过程显示在图 16.5 中。一个测试**向量**是一个二进制输入的阵列，它们被应用到需要**进行测试的器件(DUT)或测试的芯片(CUT)**。对于每个输入向量，测量响应并与期望的输出进行比较。为了充分地测试 DUT，需要不止一个的输入向量，一般都要设计一个**测试向量组**来决定芯片是否工作。因为每个测量都需要时间，希望得到一个最小的测试向量组(集)来减少决定芯片是否工作所需要的总时间。**测试向量生成**是测试的一个比较富有挑战性的方面之一。

生成测试向量的原则会因意图不同而相当不同。**功能测试**通过使芯片电路执行各种功能，并且检查它们的响应来确定这个芯片是好是坏。虽然这听起来似乎很容易，如果不加考虑地选择一组穷举的测试向量可以导致极长的测试时间。

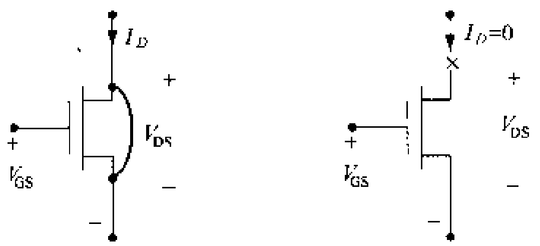
故障模拟是比较复杂的。采用芯片原型可以表示制造过程中有代表性的问题如短路、开路及坏晶体管。一旦在物理层上识别出这些故障，它们就可以用来生成专门找出这些问题的测试向量集。尽管故障模拟要求相当数量的工作，但它是很有用的，因为它找出已知问题的所在并反馈给工艺线。圆片分析可在实验室中进行以验证故障的原因且在相应的制造阶段加以改正。从长远来看，故障模拟帮助提高设计的成品率和可靠性。

16.2.1 CMOS 故障模型

物理层次的 CMOS 故障通常可用简单的等效电路来模拟。这些模型可以用在逻辑电路中，以决定一个具体的故障对一个门或逻辑单元的操作所具有的可测量的影响。FET 电路的基本故障模型比较简单。

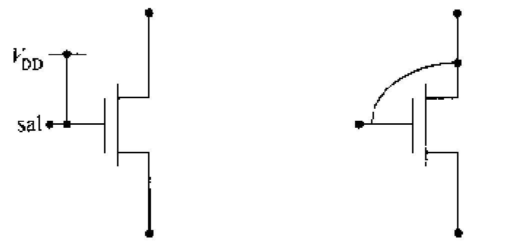
短路 FET 是指加上漏－源电压 V_{DS}时总有漏源电流导通的管子，栅对操作没有任何控制作用。这也称为**固定短路(stuck-on)**故障。固定开路或称 **stuck-off** 故障恰好相反：无论 V_{GS} 或 V_{DS}如何电流从不导通。这两种故障的电路模型多少是明显的，为了完整起见仍显示在图 16.6 中。物理上这些问题多半是由于金属化或刻蚀的问题，或是掩模标记(mask registration)出错。

两种基于逻辑的故障称为**固定 0**(struck-at-0，**sa0**)和**固定 1**(struck-at-1，**sa1**)问题。固定型故障适用于因短路或其他与工艺相关的故障而不能改变电压的连线。一种最简单情形是一个节点意外地连到电源(sa1)或地(sa0)。固定型故障的影响随发生的位置而变。例如，在 nFET 栅上的一个 sa0 故障意味着它绝不会导通，而在 pFET 栅上的 sa0 故障则使这个管子不能关断。显然，它们影响了逻辑电路的操作。相关的一组故障是栅－漏和栅－源间的短路。图 16.7 表示了这两种故障模型的例子。



(a) 短路 (b) 开路

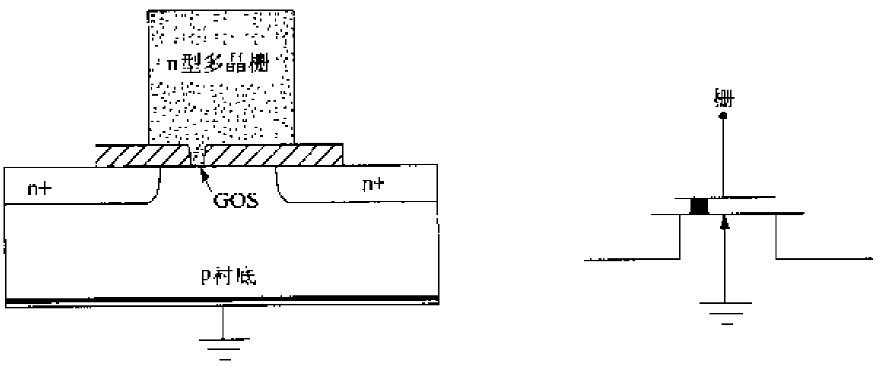
图 16.6 MOSFET 故障模型



(a) 栅 sa1 故障 (b) 栅－漏短路

图 16.7 故障模型例子

栅氧短路(GOS)是 MOSFET 特有的。它们发生在当绝缘栅氧有缺陷，使栅材料接触到衬底，如图 16.8(a)所示 nFET 的情形。假设采用 n 型掺杂的多晶栅，GOS 则在栅和 p 型衬底之间形成一个寄生的 pn 结二极管。栅压对漏电流没有任何控制作用，使该电路不能工作。栅氧短路在 nFET 和 pFET 中都会发生并且多半是由于在圆片表面的缺陷造成栅氧的不均匀生长



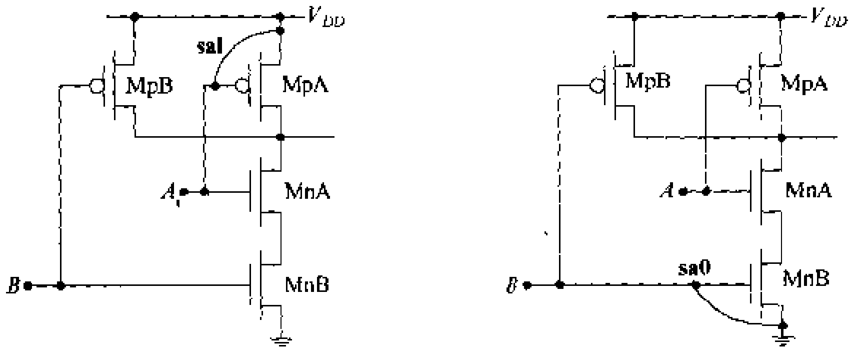
(a) 物理原因 (b) 电路表示

图 16.8 nFET 中的栅氧短路

引起的。许多 GOS 问题集中发生在圆片的一些局部地区。在这种情形下,受影响的芯片多半有许多 GOS 缺陷的 FET,因而易于发现它们。

16.2.2　门级测试

故障模型用来表示逻辑门的失效特征。构成具有不同故障的电路能够找到用于电路的一组测试向量。让我们来考察一个简单的 NAND2 门的例子,它的输入为 A 和 B。在图 16.9(a)中,在 pFET MpA 栅上的固定 1 故障使管子保持在截止状态,而 MnA 则总是导通的。图 16.9(b)的电路在 B 输入端上有一个固定 0 的故障。这阻止了 MnB 导通,且使 MpB 保持在导通状态。这两个电路代表不同的情形。它们可以用来推导为发现每个问题所需要的测试向量。图 16.10 的工作情况表提供了必要的信息。NAND 门的正常响应显示为 F。图 16.9(a)中 sa1 故障的响应表示为 F_{sa1}。由于 MpA 从不导通,这个门的输出在输入$(A,B)=(0,1)$时不能被上拉到逻辑 1。所以这个向量可以用来测试这个问题,因为它本来应当可以产生一个逻辑 1 的输出。图 16.9(b)中 sa0 故障引起输出的行为总结在 F_{sa0}这一列中。在这种情形下 MpB 总是导通所以输出固定在 1。运用一个输入向量$(A,B)=(1,1)$就可以发现这个故障。



(a) 固定1故障　　　　(b) 固定0故障

图 16.9　发生固定型故障的 NAND2 门

A	B	F	F_{sa1}	F_{sa0}
0	0	1	1	1
0	1	1	0	1
1	0	1	1	1
1	1	0	0	1

图 16.10　NAND2 的工作情况表

CMOS 测试的复杂性是由于每个电路节点都是电容性的,因此能在一个短的时间内存储电荷。如果对一个门应用一组输入测试向量,那么响应有可能受这个特点的影响。考虑在图 16.11 中 NAND 门的开路故障,它阻止 pFET MpA 导通并且应当输入组合$(A,B)=(0,1)$检测出来。然而注意到输出节点具有电容 C_{out},它不能被忽略。若将序列$(A,B)=(0,0),(0,1),(1,0),(1,1)$依次输入,那么存储电荷的存在可使该门的操作看起来是正确的。这一点可从图 16.12 的工作情况表中判断。第一个输入$(A,B)=(0,0)$得到逻辑 1 的输出,于是电容 C_{out}的两端就有电压 $V=V_{DD}$。如果快速应用下一个输入$(A,B)=(0,1)$以保证很短的测试

周期，那么输出将仍然看上去是逻辑 1，因为 C_{out} 可以保持住电荷。通过其余的输入也得到正常的结果，完全失去了发现这个故障的机会。

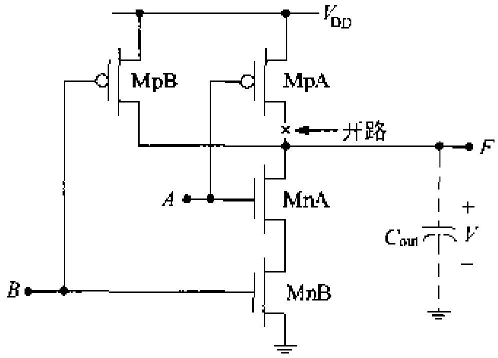


图 16.11　电荷存储对测试的影响

A	B	F	
0	0	1	← B=0使 $V=V_{DD}$
0	1	1	← 电荷维持在C上
1	0	1	← 电荷维持在C上
1	1	0	← C放电至0V

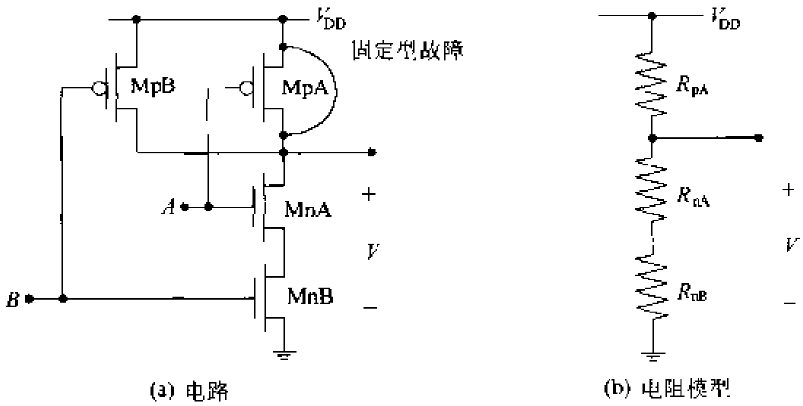
顺序↓

图 16.12　电荷存储问题的工作情况表

为了弥补这个问题，采用一个**初始化向量**，它在应用真正的测试向量之前先对这个门做些“准备”。在本例中，序列 $(A,B)=(1,1),(0,1)$ 可以发现这个故障因为初始化向量 $(A,B)=(1,1)$ 使输出放电到 0 V，于是故障就会阻止 $(A,B)=(0,1)$ 产生逻辑 1 的输出。

另一类问题来自于固定通或固定断(struck－on 和 struck－off)的故障。考虑图 16.13(a)的电路，其中 MpA 具有固定通(短路)故障。如果应用一个输入向量 $(A,B)=(1,1)$，那么 MnA 和 MnB 就会与 MpA 一起导通。这相当于图 16.13(b)所示的电阻等效模型。输出电压由分压规则得到为：

$$V=\left(\frac{R_{nA}+R_{nB}}{R_{nA}+R_{nB}+R_{pA}}\right)V_{DD} \tag{16.28}$$



(a) 电路　　(b) 电阻模型

图 16.13　NAND 门的固定型故障

因为 nFET 的电阻取决于宽长比而 R_{pA} 是由于短路，所以电压可能是一个很低的值 V，它使这个门看起来在正确工作。当两个电阻的和 $(R_{nA}+R_{nB})$ 比 R_{pA} 小时就会出现这种情形。如果 $R_{pA}\approx(R_{nA}+R_{nB})$ 那么 V 就会大约等于 V_{DD} 的一半，这也许会也许不会被检测为一个正确的值。

16.2.3 I_{DDQ}测试

对一个 CMOS 芯片加上一个电源电压可以引起一个电流 I_{DD}。当信号输入稳定时(不翻转时),可以测量到静态漏电电流 I_{DDQ}。这显示在图 16.14 中。每个芯片的设计都有一个"正常"大小的电流范围。I_{DDQ}测试的基础是认为一个不正常的漏电电流将表明芯片上存在问题。I_{DDQ}测试通常是在测试周期开始时进行的。如果一个芯片不能通过这个测试,那么它就被放弃而不再做进一步的测试。

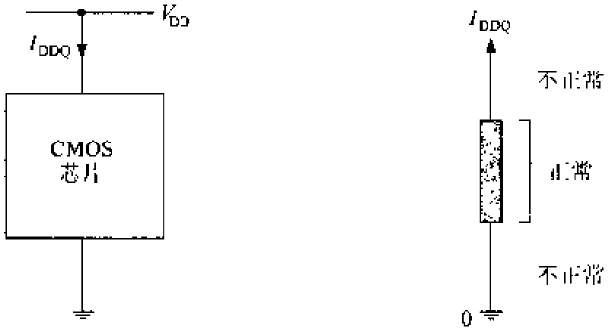


图 16.14 基本的 I_{DDQ}测试

I_{DDQ}漏电的来源显示在图 16.15 中。当把输入电压 V_{in}从小到大扫描到一个 NOT 门时,电源电流 I_{DD}的变化如图所示;峰值电流发生在中点电压即 $V_{in}=V_{out}$处。当输入稳定在逻辑 0 或逻辑 1 的电压范围时,只有静态漏电电流 I_{DDQ}流动。它包括反相偏置的 pn 结电流、亚阈值电流以及其他电流。如果测量得到漏电电流的"不正常值",那么就可以认为什么地方发生了问题。

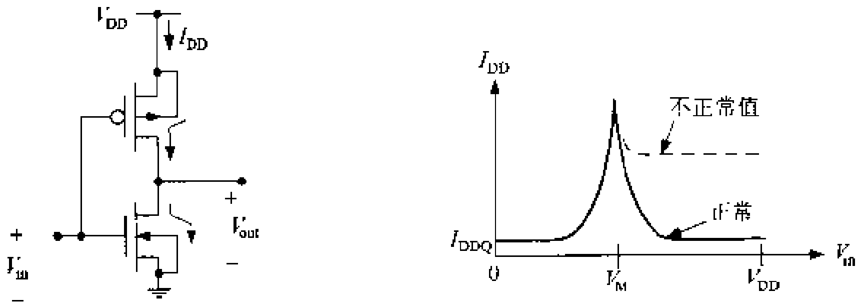


图 16.15 NOT 门中的漏电电流

图 16.16 为一个基本测量系统的组成。测试芯片模拟成与测试设备电容 C_{test}相并联。电源电压值 V_{DD}通过一个开关连到芯片,这个开关在时间 $t=0$ 时瞬间闭合。电流 I_{DD}由一个缓冲器(单位增益放大器)监测,它输出一个电压 $v_o(t)$。电流的值估计为:

$$I_{DD} \approx C\left(\frac{\Delta v_o}{\Delta t}\right) \tag{16.29}$$

式中电压在时间 Δt 内下降了一个量 Δv_o。在方程中的总电容 C 是两个电容的和:$C=C_{test}+C_{chip}$。

I_{DDQ}测试可以测出一堆 GOS(栅氧短路),因为后者往往会增加漏电值。图 16.17 显示一

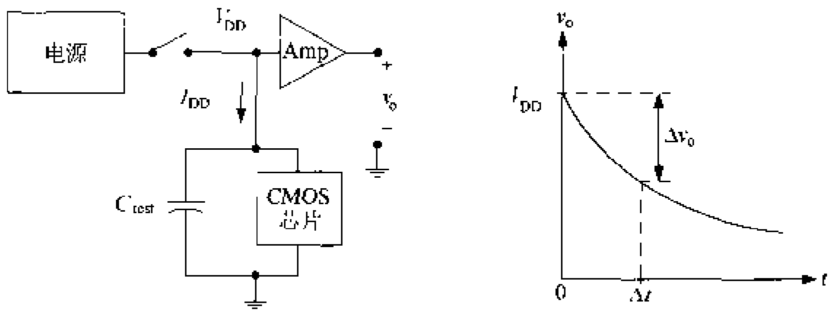


图 16.16 I_{DDQ}测量系统的组成

个有 GOS 故障的 nFET 被一个反相器电路驱动的情形。因为在 nFET 中的 GOS 故障与反偏 pn 结相同，所以当电压如图所示时在电路中就会有附加的漏电电流流动。用静态逻辑电路设计的 CMOS 芯片通常可用 I_{DDQ}方法来测试，大多数 ASIC 都属于这类设计。虽则这个技术已用于某些动态电路，但必须小心注意测试向量的形成及对测量结果的解释。

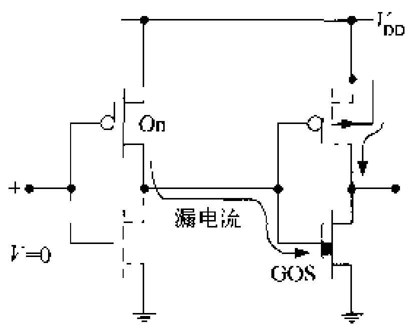


图 16.17 用 I_{DDQ}测试 GOS
(栅氧短路)

16.3 测试生成方法

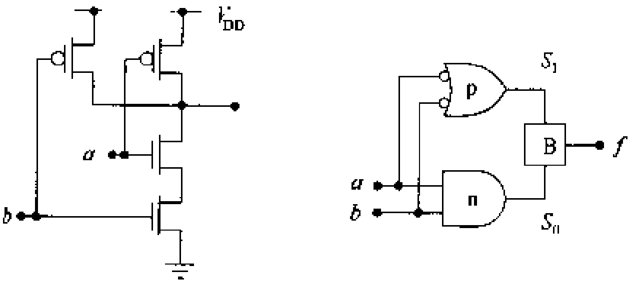
一个物理故障可以被转换成逻辑故障模型便能开发出测试向量集。已经开发了许多技术来测试采用通常电路设计类型的 CMOS VLSI 芯片。这个问题可以通过例子考察在实际电路与测试之间的关系而得到理解。

16.3.1 静态 CMOS 逻辑门

全互补 CMOS 逻辑门可用独立的 nFET 和 pFET 逻辑通路来模拟。基本的技术类似于在第 2 章中介绍的小圆圈推进方法。考虑图 16.18(a)所示的 NAND2 逻辑电路。为了建立逻辑等效电路，把串联的 nFET 看成 AND 操作而并联的 nFET 为 OR 操作；nFET 为高电平有效的器件，而 pFET 则为低电平有效的器件。它们用来构成图 16.18(b)所示的逻辑模型，它的特点是采用单独的 n 和 p 逻辑路径。n 路径的输出为 $S_0=a\cdot b$ 而 p 路径产生 $S_1=\bar{a}+\bar{b}=\bar{S}_0$。它们被送到"B 逻辑块"，产生输出 $f=f(S_0,S_1)$。

B 逻辑块的操作总结在图 16.18(c)的真值表中。对于正常的 NAND 操作，$S_0\neq S_1$即 $S_0\oplus S_1=1$。这些特点是由表中第 2 行和第 3 行表示的，它们分别生成输出 $f=1$ 或 $f=0$。如果 $S_0=S_1=0$，那么输出 $f=\mathrm{M}$，它代表一种存储状态。用电路来说明时，这意味着输出是浮空的，这是由于 FET 处于固定断状态而造成的。最后一个状况 $S_0=S_1=1$，nFET 和 pFET 均导通，所以输出被拉向两个方向。输出表示成"w0"，它代表一个"弱 0"；这是假设 nFET 下拉强度超过了 pFET 的上拉作用。如果这个假设不成立，那么输出就要改变成一个弱 1(w1)或一个中间状态。

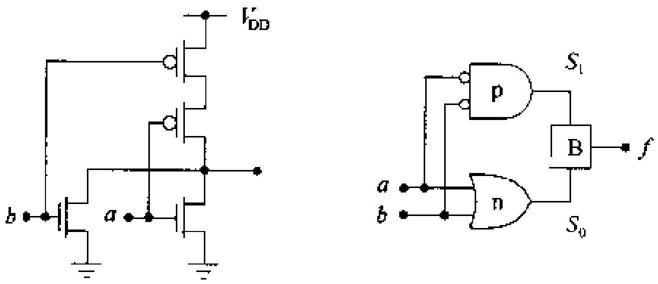
这类模型可以延伸到随意的静态逻辑门。图 16.19 为 NOR2 门的模型。串接和并联晶体管的布线决定了等效逻辑门，它的输出则被送到 B 块。注意，B 块逻辑与 NAND2 门的情形



S_0	S_1	f
0	0	M
0	1	1
1	0	0
1	1	w0

(a) 电路　　(b) 逻辑模型　　(c) B逻辑块

图 16.18　NAND2 的逻辑模型



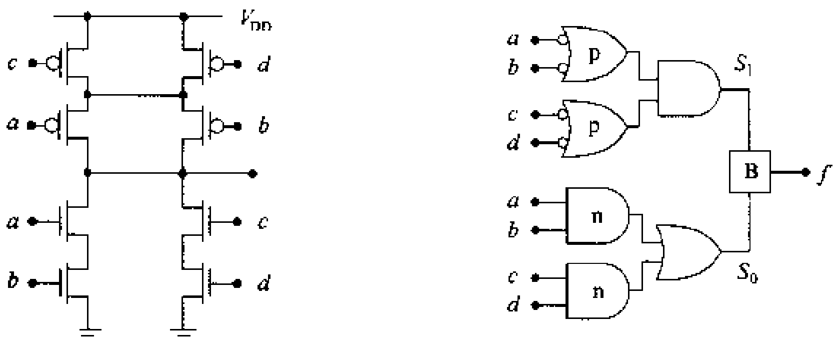
S_0	S_1	f
0	0	M
0	1	1
1	0	0
1	1	w0

(a) 电路　　(b) 逻辑模型　　(c) B逻辑块

图 16.19　NOR2 门的逻辑模型

一样。复杂逻辑门的模型是应用串联 AND 和并联 OR 的关系把管子组合起来构成的。图 16.20(a)为一个实现如下逻辑功能的 AOI 电路：

$$F = \overline{a \cdot b + c \cdot d} \tag{16.30}$$



(a) 电路　　(b) 逻辑模型

图 16.20　AOI 门的逻辑模型

nFET 逻辑等效电路表示产生 S_0的 AO 结构。p 路径把低电平有效输入送到 OA 电路中，其输出为 $S_1=\bar{S}_0$。B 逻辑块产生输出 f，若 $S_0 \neq S_1$，则它等于 F。如果故障发生在其中一条逻辑路径中时就会出现 $S_0=S_1$的情形。

16.3.2　故障的逻辑影响

一旦推导出一个等效的逻辑电路，就可以把故障模型应用到各个点中并分析在输出端的

影响。对于静态 CMOS 门的情形，16.2.2 节已经讨论了基于电路的观点。简单的逻辑级的模型提供了几条比较有用的信息。

考虑固定型故障，即固定 0 和固定 1 的影响。图 16.21 显示这些故障发生在基本逻辑门的输入端时的影响。图 16.21(a)总结了 AND 系列的响应。输出是采用合适的逻辑表计算的，并且假设所显示的输入端故障是惟一可能的问题。例如，简单的 AND 操作

$$f(a,b)=a\cdot b \tag{16.31}$$

表明在任何一个输入端的 sa0 得到 $f=0$，而在 a 处的 sa1 得到 $f=b$。采用同样的方法，基本逻辑门 OR 系列的固定型故障特性总结在图 16.21(b)中。

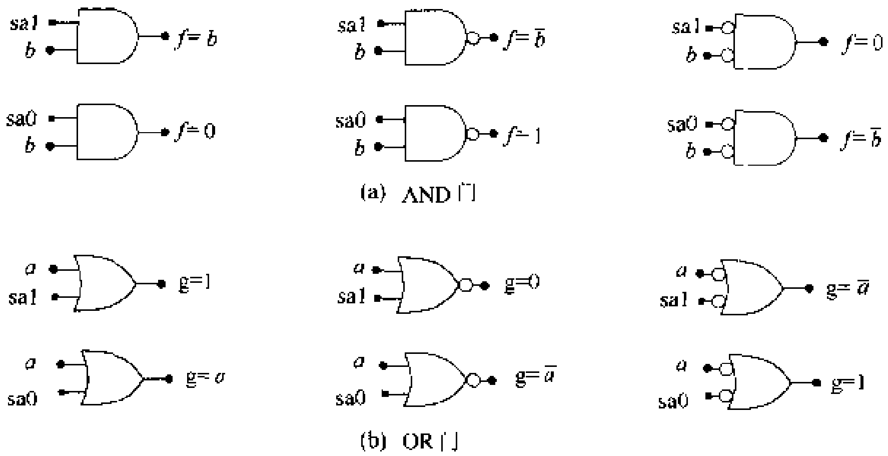


图 16.21 基本逻辑门固定型故障的影响

多个故障的情形可以采用一些巧妙的办法来处理。图 16.22 说明了某些故障简化的概念。图 16.22(a)为 sa1 故障同时发生在一个 AND 门的输入和输出时的情形。在这种情形中，输出会掩盖掉输入的故障，所以这个门左边的任何情形都可以被忽略。这称为**故障支配**，它对简化测试向量生成很有用。图 16.22(b)为等价故障。在这种情形中存在三种不同的故障，但只一个 sa0 输入 NAND2 门可以得到相同的故障行为。图 16.22(c)显示了通过**故障合并**的简化方法。由于输入和输出相同，所以任何一个门可以用短路来替代，由此得到较简单的逻辑电路。

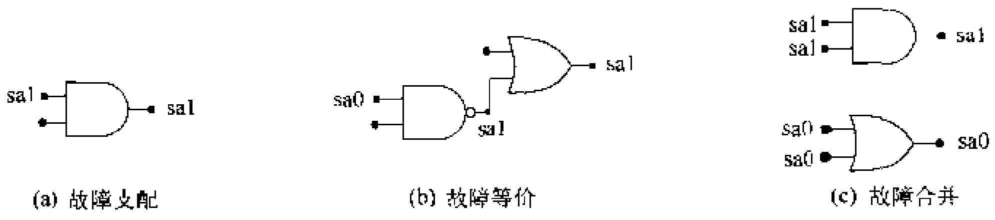


图 16.22 故障简化

这些例子表明，逻辑级的故障模型如何用来描述实际电路的缺陷。一种生成测试向量集的重要技术是把故障放在电路的不同位置上，然后计算它们的影响。以这种方式表示响应的特性，能够生成发现特定故障的测试向量。

16.3.3 路径的敏化

当被测试的门嵌入一个较大的逻辑电路中时，可以采用现有的电路来建立从故障位置到可观察输出点的特殊路径。这种技术称为**路径的敏化**，而建立这个路径的过程称为**“传播过程”**，因为故障被看成传播通过逻辑电路。

考虑图 16.23 的简单逻辑电路，它实现如下函数：

$$F = a_1 \cdot a_2 + \bar{a}_2 \cdot a_3 \tag{16.32}$$

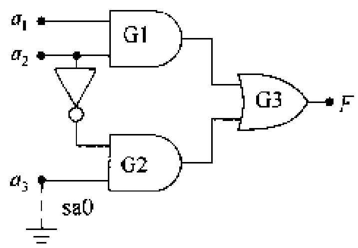


图 16.23 路径敏化例子

希望决定输入以测试在输入端 a_3处的 sa0 故障。路径敏化采用两步进行。第一步称为正向驱动，希望区分正常操作影响与故障影响。对于 sa0 故障情形，置 $a_3=1$，所以它不同于故障输入。为使这个值传播通过 AND 门 G2，必须使反相器输出为 1。与 $a_3=1$ 联合起来就得到了 G2 的输出为 1。为了把这个值传播通过 OR 门 G3，需要使 G1 的输出为 0，这就完成了正向驱动。

第二步称为反向跟踪。它运用正向驱动的结果决定检测故障所需要的输入。第一种情况是 $a_3=1$。为了保证反相器的输出为 1，必须选择 $a_2=0$。最后为了保证 G1 的输出为 0，需要使 a_1 或 a_2为 0。由于已经选择了 $a_2=0$，所以 a_1可以是 0 或 1。这就得到 sa0 故障的测试向量如下：

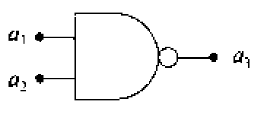
$$(a_1, a_2, a_3) = (d, 0, 1) \tag{16.33}$$

式中 d 是“无关”状态。这个简单的例子说明了这一步骤。对于单条路径并不总能得到可实现的测试向量。因此多条路径的敏化是必要的。

16.3.4 D 算法

在这个方法中，引入变量 D 来模拟一个好电路和一个有故障电路之间的差别。根据定义，$D=1$ 表示是一个好电路，而 $D=0$ 则表示存在问题。D 的反 $\bar{D}$ 以相反方式定义：$\bar{D}=0$ 是好的，$\bar{D}=1$ 是有故障的。D 算法提供了对任何可观察故障推导测试向量的技术。这一功能并不是没有复杂性的，全面的介绍远超出本书的范围。我们可以理解它的基本概念而不去涉足它的细节。

我们将考察的第一个方面是对一个门的**奇异覆盖**。这相当于在图 16.24(b)中由 NAND 例子显示的精简形式真值表中的一行，这里 d 是“无关”情形。在这一形式中有三个不同的行构成了奇异覆盖立方体。图 16.24(c)则显示了 NAND 门的基本 D 立方。根据定义，基本的 D 立方包含存在故障时在输出端产生 D 或 $\bar{D}$ 所要求的输入向量。在这里的情形中，它们相应于 $(a_1,a_2)=(0,d)$和$(d,0)$，因为它们都使输出 $a_3=1$。



(a) 逻辑门

a_1	a_2	a_3
0	d	1
d	0	1
1	1	0

(b) 奇异覆盖

a_1	a_2	a_3
0	d	D
d	0	D

(c) D立方

图 16.24 NAND2 门的奇异覆盖

一个门的传播 D 立方是为传播一个或多个输入端的 D 立方至输出端所需要的基本立方。图 16.25 为 NAND2 门的传播 D 立方。D 算法然后考察传播 D 立方的交集决定测试向量集。

D 算法在测试理论中是熟知的。这是一个非常实用的方法因为它允许运用结构化的方法来敏化多条路径。路径敏化的一个缺点是生成测试向量的过程可以变得很长和很复杂。整个时间可以通过结合**故障模拟**技术来减少。在这种方法中，应用一个测试向量然后决定什么故障可以被检测。这通常比解决逆向问题需要较少的时间。

a_1	a_2	a_3
1	D	$\overline{D}$
D	1	$\overline{D}$
D	D	$\overline{D}$
1	$\overline{D}$	D
$\overline{D}$	1	D
$\overline{D}$	$\overline{D}$	D

图 16.25 NAND2 门的传播 D 立方

16.3.5 布尔差分

测试向量生成的另一种方法是**布尔差分**。考虑图 16.26 所示的 n 个输入的电路。输出是一般的函数

$$f(a) = f(a_1, a_2, \ldots, a_n) \tag{16.34}$$

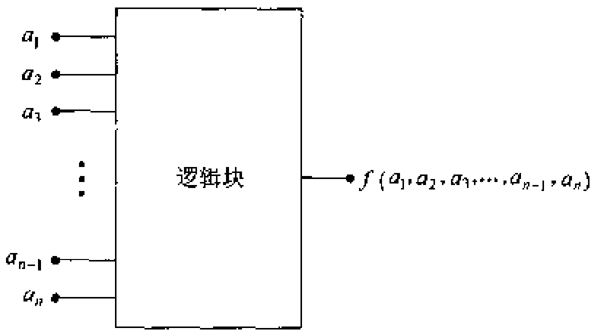


图 16.26 推导布尔差分的基本电路

让我们选择一个随意的输入 a_k 并定义

$$\begin{aligned} f_k &= f(a_1, a_2, \ldots, a_k = 1, \ldots, a_n) \\ f_{\bar{k}} &= f(a_1, a_2, \ldots, a_k = 0, \ldots, a_n) \end{aligned} \tag{16.35}$$

运用香农展开定理，我们可以把原来的函数写成：

$$\begin{aligned} f(a) &= a_k \cdot f_k + \bar{a}_k \cdot f_{\bar{k}} \\ &= a_k \cdot f_k \oplus \bar{a}_k \cdot f_{\bar{k}} \end{aligned} \tag{16.36}$$

该式给出了直接用输入 a_k 表示的表达式。

假设希望测试在 a_α 处的一个**故障**。这得到了用 $f(a)$ 表示的输出。因为这是一个故障值，我们知道如果输入 a 相同，把它与正确的输出 $f(a)$ 比较将会得到

$$f(a) \neq f_\alpha(a) \tag{16.37}$$

可以写出

$$f(a) \oplus f_\alpha(a) = 1 \tag{16.38}$$

并且定义一个测试参数

$$t_\alpha = f(a) \oplus f_\alpha(a) \tag{16.39}$$

使 $t_\alpha = 1$ 表示存在一个故障。

假设在 a_k 处存在一个 sa0 故障。对这种情形的测试参数为

$$\begin{aligned} t_\alpha &= f(a) \oplus f_{\bar{k}}(a) \\ &= [a_k \cdot f_k \oplus \bar{a}_k \cdot f_{\bar{k}}] \oplus f_{\bar{k}} \\ &= a_k \cdot f_k \oplus (\bar{a}_k + 1) f_{\bar{k}} \\ &= a_k \cdot f_k \oplus a_k f_{\bar{k}}(a) \end{aligned} \tag{16.40}$$

或

$$t_\alpha = a_k \cdot (f_k \oplus f_{\bar{k}}) \tag{16.41}$$

布尔差分定义为

$$\frac{\partial f}{\partial a_k} = f_k \oplus f_k \tag{16.42}$$

由此得到

$$t_\alpha = a_k \cdot \left(\frac{\partial f}{\partial a_k}\right) \tag{16.43}$$

可以看到

$$\frac{\partial f}{\partial a_k} = 1 \qquad \text{iff } f_k \neq f_{\bar{k}} \tag{16.44}$$

这意味着改变 a_k 也改变输出，所以故障是可以观察到的。对在 a_k 处测试 sa0 给输入赋以固定值故障的反。因而测试向量必定满足如下条件：

$$a_k \cdot \left(\frac{\partial f}{\partial a_k}\right) = 1 \tag{16.45}$$

反之，如果希望测试的是在 a_k 处的 sa1 故障，就使如下条件成立

$$\bar{a}_k \cdot \left(\frac{\partial f}{\partial a_k}\right) = 1 \tag{16.46}$$

来决定测试向量。

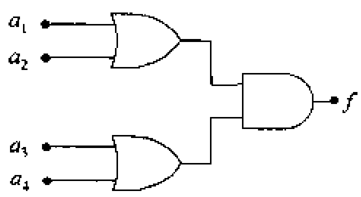


图 16.27　应用布尔差分的例子

作为如何运用这一方法的一个例子，考虑图 16.27 中的简单的 OA 电路。输出函数为

$$f(a) = (a_1 + a_2) \cdot (a_3 + a_4) \tag{16.47}$$

假设希望检测在 a_3 处的故障，要求

$$\begin{aligned} f_{\bar{3}} &= (a_1 + a_2) \cdot (a_4) \\ f_3 &= (a_1 + a_2) \cdot (1) \end{aligned} \tag{16.48}$$

所以布尔差分为

$$\begin{aligned}\frac{\partial f}{\partial a_3} &= f_{\bar{3}} \oplus f_3 \\ &= (a_1 + a_2) \cdot (a_4) \oplus (a_1 + a_2) \\ &= (a_1 + a_2) \cdot \bar{a}_4\end{aligned} \tag{16.49}$$

对于在 a_3处的 sa0 故障,采用如下条件

$$a_3 \cdot \left(\frac{\partial f}{\partial a_3}\right) = 1 \tag{16.50}$$

于是得到方程

$$a_3 \cdot (a_1 + a_2) \cdot \bar{a}_4 = 1 \tag{16.51}$$

满足这个关系的测试向量为

$$(a_1 a_1 a_3 a_4) = (1d10) \text{ 或 } (d110) \tag{16.52}$$

式中 d 是"无关"输入。类似地,在 a_3处的 sa1 故障得出如下条件

$$\bar{a}_3 \cdot (a_1 + a_2) \cdot \bar{a}_4 = 1 \tag{16.53}$$

及测试向量为:

$$(a_1 a_1 a_3 a_4) = (1d00) \text{ 或 } (d100) \tag{16.54}$$

布尔差分技术也可用来生成一个逻辑电路内部节点的测试向量。

16.4 小结

可靠性和测试是现代 VLSI 系统中的关键方面。在这个简短的介绍中只是"触及了表面",但力图说明一些主要的问题和解决办法。

随着 VLSI 系统复杂性的增加,测试变得越来越困难。可靠性继续是一个需要考虑的问题,因为用户都希望他们的系统将能永远工作下去。有兴趣花时间进一步研究的读者将会发现这可能是一条一生的事业道路,它对以后每一代 VLSI 的发展都是非常重要的。

16.5 参考资料

[1] Harry Bleeker, Peter van den Eijnden, and Frans de Jong, **Boundary-Scan Test**, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.

[2] Niraj K. Jha and Sandip Kunda, **Testing and Reliable Design of CMOS Circuits**, Kluwer Academic Publishers, Norwell, MA, 1990.

[3] Arthur B. Glaser and Gerald E. Subak-Sharpe, **Integrated Circuit Engineering**, Addison-Wesley, Reading, MA, 1977.

[4] Ravi K. Gulati and Charles F. Hawkins (eds), **I_{DDQ} Testing of VLSI Circuits**, Kluwer Academic Publishers, Norwell, MA, 1993.

[5] Kenneth P. Parker, **The Boundary-Scan Handbook**, Kluwer Academic

Publishers, Norwell, MA, 1992.

[6] Paul A. Tobias and David Trindade, **Applied Reliability**, Van Nostrand Reinhold, New York, 1986.

[7] Michael John Sebastian Smith, **Application-Specific Integrated Circuits**, Addison-Wesley Longman, Reading, MA, 1997.

[8] Neil H.E. Weste and Kamran Eshraghian, **Principles of CMOS VLSI Design**, 2nd ed., Addison-Wesley, Reading, MA, 1993.